VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY
INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

VYUŽITÍ UMĚLÉ INTELIGENCE VE VIBRODIAGNOSTICE
UTILIZATION OF ARTIFICIAL INTELLIGENCE IN VIBRODIAGNOSTICS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Petra Dočekalová

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Daniel Zuth, Ph.D.

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Studentka:	**Bc. Petra Dočekalová**
Studijní program:	Strojní inženýrství
Studijní obor:	Kvalita, spolehlivost a bezpečnost
Vedoucí práce:	**Ing. Daniel Zuth, Ph.D.**
Akademický rok:	2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Využití umělé inteligence ve vibrodiagnostice

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat možností využití metod umělé inteligence pro vyhodnocení poruch strojního zařízení. Důležitou částí práce bude přehled a popis použitelných metod a následně aplikace některých metod na vzorku naměřených dat. Aplikace může být řešena v libovolném programovém prostředí jako Matlab, Octave, Knime atd.

Cíle diplomové práce:

Rešerše v oblasti metod uměle inteligence.
Vybrat a popsat vhodné programovací prostředí (knihovny).
Vybrat a popsat vhodné metody klasifikace dat (nejméně 3).
Popsat zdrojová data pro následně zpracování.
Vytvořit aplikaci ve vybraném SW prostředí.
Vyhodnotit úspěšnost klasifikace včetně vizualizace.
Dokumentace vytvořeného řešení.

Seznam doporučené literatury:

DANIŠ, Stanislav. Základy programování v prostředí Octave a Matlab. Praha: Matfyzpress, 2009. ISBN 978-80-7378-082-1.

VONDRÁK, Ivo. Umělá inteligence a neuronové sítě. 3. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2009. ISBN 978-80-248-1981-5.

KNIME | Open for Innovation. KNIME | Open for Innovation [online]. Copyright © Copyright 2019 KNIME AG. All Rights Reserved. [cit. 18.09.2019]. Dostupné z: https://www.knime.com/

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

ABSTRAKT

Diplomová práce pojednává o strojovém učení, expertních systémech, fuzzy logice, genetických algoritmech, neuronových sítích a teorii chaosu, které spadají do kategorie umělé inteligence. Cílem práce je popsat a implementovat tři různé klasifikační metody, podle kterých bude zpracován soubor dat. Pro aplikaci dat bylo zvoleno SW prostředí GNU Octave z licenčních důvodů. Dále vyhodnotit úspěšnost klasifikace dat včetně vizualizace. Pro srovnání jsou použity tři různé klasifikační metody, abychom mohli zpracovaná data mezi sebou porovnávat.

ABSTRACT

The diploma thesis deals with machine learning, expert systems, fuzzy logic, genetic algorithms, neural networks and chaos theory, which fall into the category of artificial intelligence. The aim of this work is to describe and implement three different classification methods, according to which the data set will be processed. The GNU Octave software environment was chosen for the data application for licensing reasons. Further evaluate the success of data classification, including visualization. Three different classification methods are used for comparison, so that we can compare the processed data with each other.

KLÍČOVÁ SLOVA

Umělá inteligence, fuzzy logika, genetické algoritmy, klasifikační metody, rozhodovací stromy, K-NN, SVM, neuronové sítě, confusion matrix, GNU Octave

KEYWORDS

Artificial intelligence, fuzzy logic, genetic algorithms, classification methods, decision trees, K-NN,SVM, neural networks, confusion matrix, GNU Octave

BIBLIOGRAFICKÁ CITACE

DOČEKALOVÁ, P. Využití umělé inteligence ve vibrodiagnostice, Brno, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2021, 70 s., Vedoucí diplomové práce Ing. Daniel Zuth, Ph.D.

PODĚKOVÁNÍ

Chtěla bych tímto poděkovat Ing. Danielovi Zuthovi, Ph.D. Za cenné rady a odborné vedení při vypracování mé diplomové práce. Zároveň bych chtěla poděkovat mé rodině a nejbližším za podporu a trpělivost.

ČESTNÉ PROHLÁŠ ENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracovala jsem ji samostatně pod vedením Ing. Daniela Zutha, Ph.D a s použitím literatury uvedené v seznamu.

V Brně dne 20.05.2021 ……

Dočekalová Petra

OBSAH

1 ÚVOD

V dnešní době se snaží firmy efektivně využívat firemní data. Data z měření nebo z různých firemních databází se využívají pro analýzy a vizualizaci dat, které umožňují zaměstnancům ve firmě lépe pochopit danou problematiku. Na trhu je vysoká konkurence, proto se firmy snaží být o krok napřed před konkurencí. Většina lidských činností ve firmách je nahrazována stroji a umělou inteligencí, přesně tak i v technické diagnostice. K celkové automatizaci dochází především z důvodu vyšší efektivity a ke snížení ekonomických nákladů.

Cílem této diplomové práce je vytvoření klasifikačních modelů pro zpracování dat pomocí umělé inteligence v oblasti vibrodiagnostiky. Důležitým pojmem v této práci je strojové učení, které se dokáže ze získaných dat učit a zároveň z nich vyvozovat závěry. Jedná se především o nalezení spojitosti (podobnosti) mezi daty, díky kterým jsme schopni v budoucnu definovat aspekty, která mohou stroj v budoucnu ovlivnit a která ne.

Teoretická část popisuje umělou inteligenci a její odvětví, do kterého spadají genetické algoritmy, fuzzy logika, neuronové sítě, teorie chaosu a především strojové učení. Dále v teoretické části jsou popsány čtyři klasifikační metody. Z toho jsou tři metody poté implementovány na neroztříděný soubor dat. Poslední kapitola popisuje SW prostředí.

V praktické části jsou nejprve popsána zdrojová data a jejich zpracování pro následnou práci s klasifikačními modely. Předpříprava a klasifikace dat probíhá v prostředí GNU Octave. Jedná se o stanovení poruch rotačního stroje z měřeného vzorku v časové oblasti. Použití klasifikátorů může urychlit řešení opakujících se poruch na strojích a omezit prostoje ve výrobě.

Na konci práce nechybí celkové zhodnocení všech klasifikačních modelů.

2 UMĚLÁ INTELIGENCE

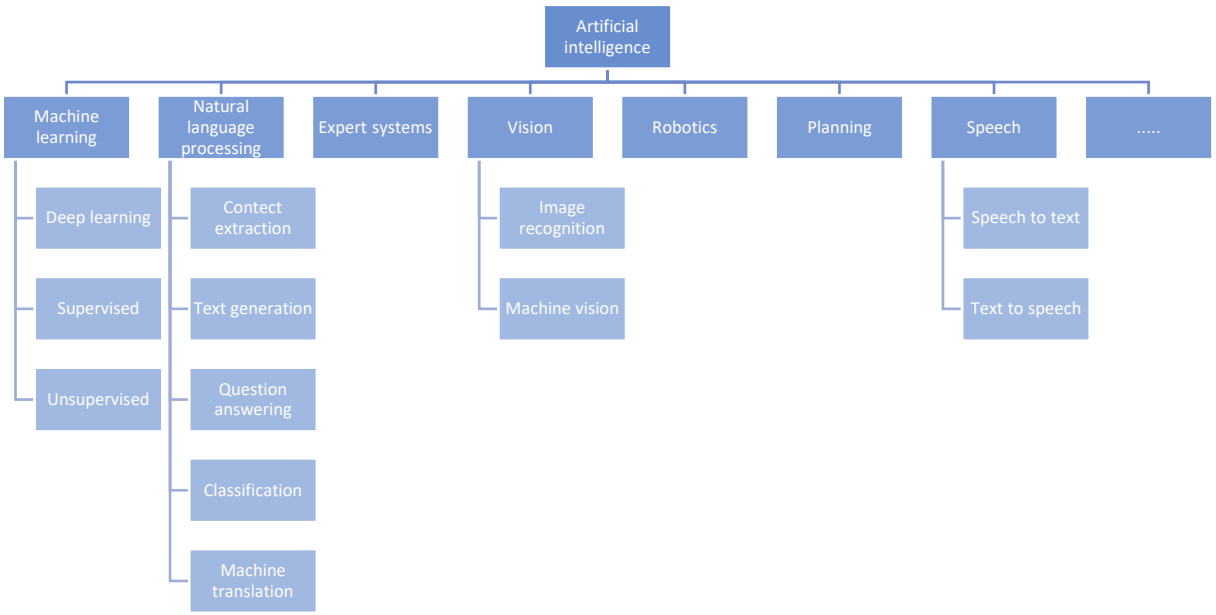
Pod názvem umělá inteligence si můžeme představit schopnosti určitých počítačových programů, které se snaží napodobit inteligenci člověka. Nejde jednoznačně definovat. Důvodem je multioborové zaměření této vědní disciplíny. [2]

Umělá inteligence se projevuje v několika různých podobách, jako jsou: [27]

- Chatovací roboti využívají UI pro rychlejší pochopení problémů zákazníků a poskytování odpovědí
- Robotičtí asistenti provádějí analýzu důležitých dat z velkých datových souborů s volným textem, aby zlepšili plánování.
- Moduly doporučení poskytují doporučení televizních pořadů na základě statistik zájmů diváků.

Umělá inteligence se více zabývá procesy a schopnostmi vyspělého myšlení a analýzy dat než konkrétních formátů a funkcí. Hlavním cílem umělé inteligence je zvýšit schopnost a užitečnost člověka a umožnit řešit problémy, které byli dříve neřešitelné a nepochopitelné.

Umělou inteligenci dělíme na několik dalších podoblastí, které jsou:



Obr. 1) Rozdělení UI [8]

2.1 Historie umělé inteligence

Jedná se o jednu z nejrychleji vyvíjejících vědních a technických disciplín v historii. Za růst této vědní disciplíny může několik faktorů, jako např. zvyšování požadavků v oblasti automatizovaného řízení, průzkumu nedosažitelných míst a mnoho dalších činností, kde je přítomnost člověka z technických nebo ze zdravotních důvodů vyloučená.[2]

Významné milníky v umělé inteligenci: [1]

Obr. 2) Mezníky v UI

1941 - poprvé byl použit elektronický počítač v Německu. Počítač byl tak obrovský, že byl chlazen leteckými motory.

50.léta- Začali se vyvíjet umělé neuronové sítě jako výsledek snah napodobit způsob, jakým způsobem pracuje lidský mozek.

1956- John McCarthy definoval pojem umělá inteligence jako obor informatiky zabývající se tvorbou strojů vykazující inteligentního chování.

60. léta- Vznikalo mnoho programů, např. STUDENT, který dokázal řešit algebraické problémy, nebo program SIR, který rozuměl jednoduchým anglických větám.

1961- V továrně firmy General Motors byl instalován první průmyslový robot.

1963- MIT dostala vysoký grant pod ministerstva obrany USA, aby byla zajištěna technologická výhoda proti Sovětskému svazu. Projekt vedl k velkému rozvoji v oblasti UI.

70. léta- Příchod expertních systémů. V těchto letech vzniklo velké množství nových metod, např. teorie o strojovém vidění, jazyk PROLOG

80. léta- Byl zaznamenán zrychlený vývoj UI. Zvýšila se poptávka po expertních systémech.

1997- Počítač IBM Deep Blue porazil slavného ruského šachistu a mistra světa Garriho Kasparova

1999- Na japonský trh byl uveden robotický pes AIbo. Během 20 minut byly všechny kusy prodány.

2004- Roboti Opportunity a Spirit úřadu NASA hledali na Marsu vodu.

2009- společnost Google vyvíjí auto, které řídí samo.

2016- Superpočítač společnosti Google ALphaGo porazil poprvé v historii mistra světa v asijské deskové hře Go, když zdolal Jihokorejce I Se-dola.

2.2 Umělá inteligence v diagnostice

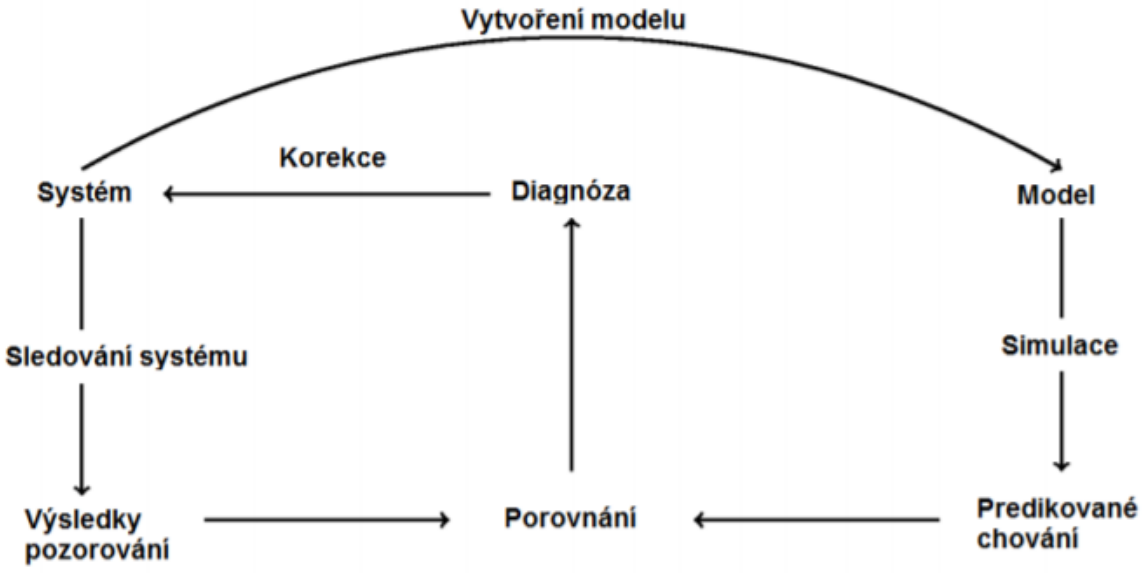
Nastávají situace, kdy základní diagnostické algoritmy nedosahují požadované úrovně citlivosti, přesnosti a jednoznačnosti. Proto se stále více využívá prostředků umělé inteligence a to především expertních systémů, neuronových sítí, fuzzy logiky, evolučních systémů a jejich kombinacích. [3,36]

Diagnostika z hlediska umělé inteligence, lze rozdělit do tří kritérií:

- Modelování systémů
- Expertní posouzení
- Kombinovaný přístup

Modelování systému [3,35]

Modelování je proces, kde se sestavuje matematický popis určitého fyzikálního systému. Matematický popis charakterizuje vstupní a výstupní chování pozorovaného systému. Popsaný systém může být díky popisu kontrolován, simulován, predikován a případně vylepšen. Tvorbou spolehlivých modelů se zabývá systémové modelování. Proces systémového modelování může vypadat tímto způsobem viz na Obr.2.



Obr. 3) Proces systémového modelování [35]

K popisu fyzikálních systémů se používají matematické rovnice a vztahy, které daný systém mohou popisovat jak kvantitativně, tak i kvalitativně. Tento vytvořený popis systému se označuje jako matematický model.

Většina fyzikálních systémů je velmi složitě matematicky přesně a jasně popsatelná. Důvodem je složitost struktury systému, nelinearity, nejasnosti atd. Proto se často využívá aproximačního modelování.

Pro aproximační modelování se využívá fuzzy logika. Využívá se především vágní a složité fyzikální systémy, které nelze jednoduše matematicky popsat. Fuzzy logika využívá pro aproximaci dat funkce příslušnosti. Proces tvorby fuzzy modelů se nazývá fuzzy modeling. Využití fuzzy systému není vždy jednoduché. Volba vhodných funkcí příslušností a sestavení vhodných pravidel je časté zkoušení a upravování parametrů.

Expertní posouzení

Diagnostika pomocí expertního posouzení je založena na zkušenostech a znalostech nad daným systémem. Cílem je vytvoření zobrazení, které vychází ze sdružování informací k odpovídající diagnóze.

Je důležité data správně interpretovat do počítačového jazyka nebo pozorovat chování diagnostikovaného systému a správně vyhodnotit.[3]

2.2.1 Vibrodiagnostika

Vibrodiagnostika je nejpoužívanější metoda pro sledování technického stavu zařízení. Jedná se o bezdemontážní diagnostickou metodu, která je prováděna při práci na stroji a hodnotí mechanické kmitání Toto kmitání je měřeno na pohyblivých a nepohyblivých částech diagnostického stroje. [11]

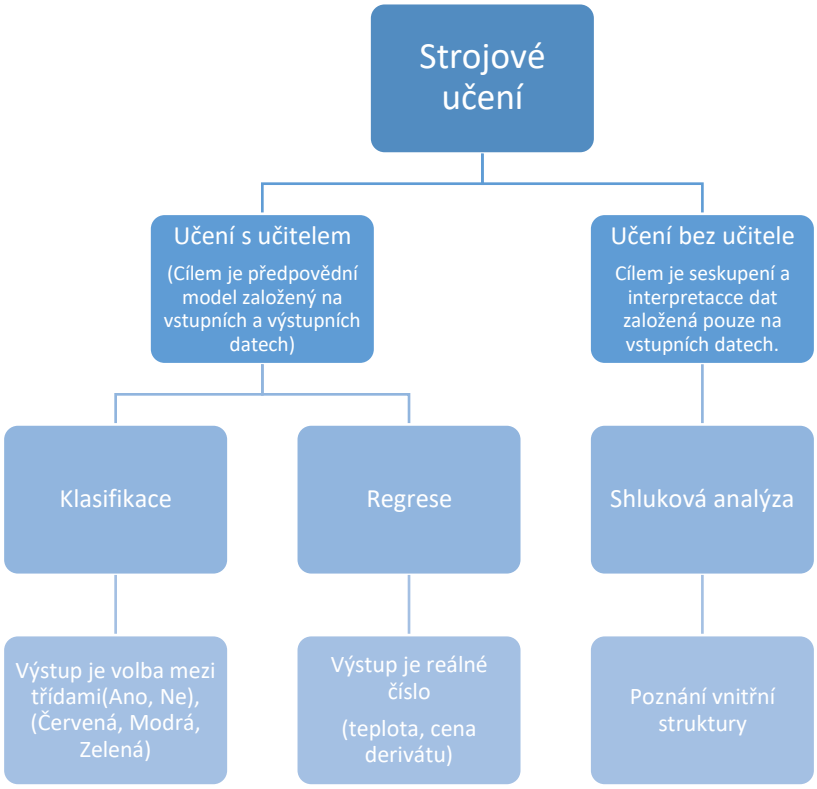
Vibrace vznikají v důsledku mechanického kmitání stroje. Mohou být způsobeny technickým stavem ložisek, nesouosostí, nevývahou, stykem třecích ploch apod. [11]

U strojů se nejčastěji vyskytují složené vibrace s vibracemi náhodnými tzv. šumem, který při náhodném zpracování signálů často překrývá signál obsahující podstatné informace. [11]

Při využití vibrodiagnostiky je vyhodnocováno několik signálů, které charakterizují chvění stroje. Senzory vibrací je možné rozdělit do dvou skupin. První skupinou jsou tzv. seismická zařízení, a nebo také senzory absolutních vibrací, např. akcelerometr. Tyto senzory jsou připevňovány na konstrukci strojů, kde je pohyb prvků vztahován k pevnému bodu v gravitačním poli Země. Druhou skupinou jsou snímače relativní výchylky, které vyhodnocují danou veličinu na základě jiné části stroje. Volba snímačů závisí na měřené veličině (výchylka, rychlost, zrychlení) vibrací a zda chceme měřit absolutní nebo relativní vibrace. [11]

2.3 Strojové učení

Strojové učení (Machine learning) je součástí umělé inteligence. Jedná se o proces, který využívá matematické modely dat, pomocí kterých se vývojové systémy učí a zvyšují svoji výkonnost na základě dat, se kterými pracují. Využívá algoritmy k identifikaci vzorů v datech a tyto vzory se používají k vytvoření datového modelu, který definuje předpovědi. Strojové učení je všude kolem nás, např. komunikace s bankou, nákup přes internet nebo využívání sociálních sítí a zde vstupují do hry algoritmy strojového učení, jejichž účelem je zefektivnit, zjednodušit a zabezpečit naše aktivity. [6,7]

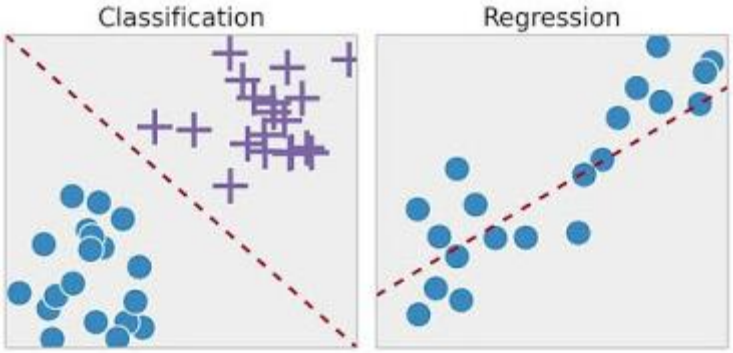


Obr. 4) Rozdělení strojového učení podle typu úlohy [41]

2.3.1 Učení s učitelem

Nejběžnější způsob strojového učení, který dnes funguje efektivně [8]

- Klasifikace- rozděluje vstupní data do dvou nebo několika tříd
- Regrese- odhaduje číselnou hodnot výstupu podle vstupu

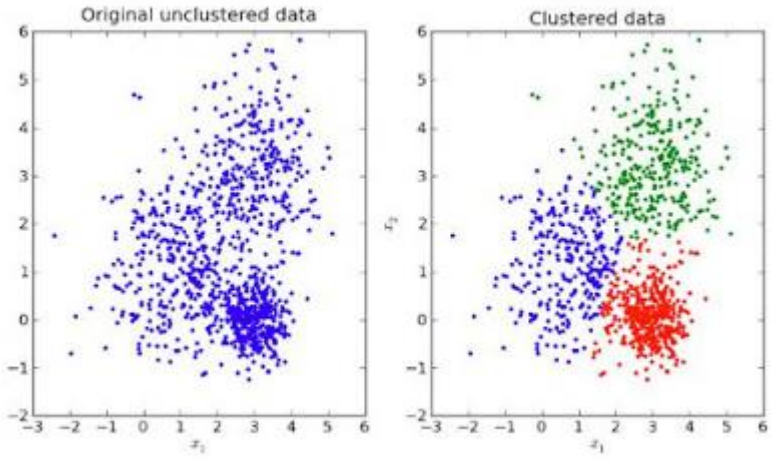


Obr. 5) Učení s učitelem [8]

2.3.2 Učení bez učitele

Tento algoritmus trénuje na vstupních datech, kde nezná rozřazení do jednotlivých skupin. Neexistuje žádné kritérium pro správnost hledané transformace vstupních dat. Data se dělí na základě podobnosti, tzv. shlukování dat. [31]

Shlukování-zařazuje objekty do skupin s podobnými vlastnostmi: na základě předkládaných vzorků vstupních dat provádí jejich třídění do skupin, aniž by byla možnost posoudit správnost zatřídění. [30]



Obr. 6) Učení bez učitele [8]

2.3.3 Deep learning (hluboké učení)

Využívá algoritmů (především neuronových sítí) s velkým počtem vrstev popisující data. Hloubka modelu je definována počtem vrstev, které jsou složeny z umělých neuronů. Vrstvy jsou navzájem propojeny, že výstup z jedné je vstupem do následující vrstvy. Hloubka vrstev je obvykle v řádech desítek a více vrstev. [14]

Základním algoritmem pro hluboké učení (vícevrstvé perceptrony) je algoritmus zpětného šíření chyb. [14]

2.4 Zpracování přirozeného jazyka

Zpracování přirozeného jazyka je v překladu natural language processing (NLP). Jedná se o interdisciplinární obor, který zahrnuje lingvistiku, počítačovou vědu a kognitivní psychologii. NLP se zabývá teoreticky výpočetní technikou pro analýzu a reprezentaci textových dat. [15]

"Texty jsou zpracovávány v rámci jedné nebo více úrovní lingvistických analýz (lexikální, sémantická, morfologická apod.) za účelem napodobit lidské zpracování jazyka pro různorodé aplikace. Vzhledem k faktu, že NLP usiluje o napodobení člověka, je patřičné na ni nahlížet jako na disciplínu AI." (Rathouz, 2017)

Umožňuje strojům číst a porozumět lidskému jazyku. Aplikace, které zpracovávají přirozený jazyk zahrnují:

- získávání informací IR -je činnost získávání zdrojů informačního systému, které jsou relevantní pro informační potřebu ze sbírky těchto zdrojů. IR systém poskytuje přístup k knihám, časopisům a jiným dokumentům; tyto dokumenty spravuje a ukládá. [17]
- dolování textu nebo-li těžba dat. Je proces odvozování vysoce kvalitních informací z textu. [18]
- odpovídání na otázky QA- je část informatiky v oblasti získávání informací a zpracování přirozeného jazyky (NLP), která se zabývá stavebními systémy, které automaticky zodpovídají otázky položené člověkem v přirozeném jazyce. [19]
- strojový překlad MT- jedná se podoblast počítačové lingvistiky, která používá software k překladu textu nebo řeč z jednoho jazyka do druhého. [20]

2.5 Expertní systém

Definice expertního systému zní: *„Expertní systémy jsou počítačové programy, simulující rozhodovací činnost experta při řešení složitých úloh a využívající vhodně zakódovaných, explicitně vyjádřených znalostí, převzatých od experta, s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni experta."* [10]

Struktura expertního systému se skládá z těchto základních složek: [32]

- bázi znalostí
- inferenční mechanismus
- I/O rozhraní (uživatelské, vývojové, vazby na jiné systémy)
- vysvětlovací modul
- modul pro získávání znalostí

Expertní systém je možné použít pro řešení problému pouze tehdy, jestli jsou splněny následující dvě podmínky: [32]

- Problém, který je složitý svým rozsahem nebo neurčitostí vztahů pro nějž exaktní metoda řešení není k dispozici, nebo není schopna poskytnout řešení v požadované době.
- Efekty plynoucí z použití expertního systému musí převyšovat vynaložené náklady. Mělo by se jednat o problém s opakovanou potřebou řešení na značnými finančními dopady, pro niž jsou lidští experti drazí nebo omezeně dostupní.

Expertní systémy můžeme hodnotit podle několik kritérií. Podle obsahu báze můžeme expertní systémy rozdělit na: [32]

- problémově orientované - báze, která obsahuje znalosti pouze určité domény
- prázdné - báze znalostí je prázdná

Podle způsobu řešení problému dělíme expertní systémy, na: [32]

- Diagnostické - hlavním úkolem je určit, která hypotéza z předem definované konečné množiny cílových hypotéz nejlépe koresponduje s daty týkajícími se daného konkrétního případu.
- Plánovací- vyhodnocuje úlohy, kde je znám počáteční stav a cíl řešení. Je nutné využít data z konkrétního řešeného případu a nalézt posloupnost kroků, kterými lze cíle dosáhnout.

2.6 Vnímání

Vnímání stroje je schopnost počítačového systému vykládat data podle toho, jak lidé používají své smysly, které vztahují na svět kolem sebe. Základním úkolem počítače je přijímat a reagovat na jejich prostředí, které je připojeno přes hardware. Nejprve byly vstupy omezeny na klávesnici a myš, ale rozvoj technologií umožňuje počítačům přijímat smyslové vjemy podobným způsobem jako člověk. [25]

2.7 Robotika

Cílem robotiky je navrhnout a vyrobit inteligentní stroje, které mají pomoci nebo nahradit člověka v každodenních pracovních povinnostech.

Ve firmách se zavádí ve výrobních prostorách tzv. automatizace (robotizace), která vede ke snižování lidské činnosti ve výrobě, zkracování výrobních časů, zvyšování výkonů a produktivity práce. [26]

2.8 Speech

Jedná se o proces, který modeluje přirozený jazyk. Daný proces vyžaduje změnu jednotek textu na jednotky řeči pro zvukovou prezentaci. Technologie bere mluvená slova a snaží se je přepsat v text. Převod textu na řeč je vykreslován pomoci zvukových výstupů z digitálního textu, který pomáhá lidem, kteří nejsou schopni číst nebo k jiným účelům. [27]

2.9 Plánování

Plánování AI je někdy označováno jako automatické plánování a rozvrhování. Jedná se o odvětví umělé inteligence, které se týká realizací strategií nebo akčních sekvencí za pomoci inteligentních agentů, autonomních robotů a bezpilotních vozidel. Řešení problémů je složitá operace, kde data musí být objevena a optimalizována ve více rozměrném prostoru. Plánování souvisí s teorií rozhodování. [20]

Ve známém prostředí s dostupnými modely lze plánovat offline. Řešení lze najít i před samostatným provedením. V neznámém prostředí musí být většinou strategie revidována online. Modely musí být přizpůsobeny. Pro řešení jsou obvykle používány iterační procesy pokusů a omylů, které jsou běžně pozorovatelné v AI. Zahrnují dynamické plánování, učení posilování a kombinatorická optimalizace. K popisu plánování se jazyky nazývají akční jazyky. [20]

Dynamické plánování nebo-li programování je metoda matematické optimalizace. Zakladatelem metody je Richard Bellman, který ji založil v 50. letech 20.století. Metoda našla uplatnění v mnoha oborech, od leteckého inženýrství až po ekonomiku. [21]

Učení zesílení/ posílení (RL) je odvětví strojového učení, které se zabývá tím, jak by softwaroví agenti měli podnikat kroky v prostředí, aby maximalizovali určitou představu o kumulativní odměně. Jedná se o velmi obecný problém, který je řešen v mnoha jiných oborech, jako např. teorie her, operační výzkum, optimalizace simulační báze, multiagentní systémy, inteligentní roj, statistika a genetický algoritmus.[22]

Kombinatorická optimalizace je téma v operačním výzkumu, aplikované matematice a teoretické informatice, které se zabývá nalezením optimálního objektu konečné sady objektů.[23]

aplikace, které optimalizaci zahrnují, jsou:

- optimální doručování balíků
- vypracování nejlepšího přidělování pracovních míst lidem
- logistika
- optimalizace dodavatelského řetězce

2.10 Stavový prostor a jeho prohledávání

Slouží k jednoduššímu uchopení a vymezení problému, který se nachází v nalezení posloupností jednotlivých akcí od počátku stavu úlohy (zadání) až k požadovanému řešení. Je tvořen množinou proměnných, které mohou nabývat hodnot pouze z určitého intervalu. Tento interval hodnot určuje celkový stavový prostor.

Stavový prostor je definovaný vztahem $SP = (S, \Phi)$. S je konečnou neprázdnou množinou stavů (v obecném případě i nekonečnou). Φ představuje neprázdnou konečnou množinu operátoru, kde každý operátor je parciální funkcí na stavovém prostoru (nemusí být definován všude). [13]

$$\alpha: S \rightarrow S \tag{1}$$

Stavový prostor je: [12]

- Konečný, např. stavový prostor vyhozené koruny má pouze dva stavy-> při dopadu padne panna nebo orel. Má omezený počet možných stavů.
- spočitatelný, např. vržení hrací kostky-> 6 stavů. Ke každému stavu jsme schopni přiřadit jedno přirozené číslo.
- nekonečný, např. planeta obíhající okolo slunce. Proměnné jsou reálná čísla, ale které nejsou spočitatelná, jako souřadnice planety.

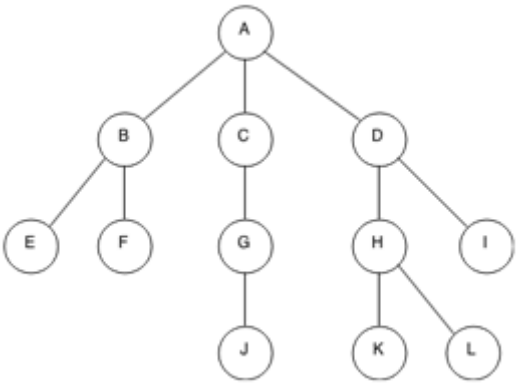
2.10.1 Prohledávání stavového prostoru

Stavový prostor můžeme popsat orientovaným grafem G (stavovým grafem, stromem).

$G = (V;E)$,

V= vrcholy(uzly) popisují stav

E= hrany reprezentují přechod mezi stavy



Obr. 7) Stavový graf, strom [13]

Způsob prohledávání stavového prostoru: [13]

1. Neinformovaní prohledávání (dnes)
    a) Do šířky
    b) Do hloubky
    c) IDFS do hloubky s omezenou max.hloubkou
    d) Dvousměrné prohledávání od poč. stavu i od cíle

2. Informované prohledávání (příště) - prohledávání na základě další informace, odhadu vzdálenosti od cíle
    a) Paprskové
    b) gradientní
    c) uspořádané
    d) A*

2.11 Genetické algoritmy

Genetické algoritmy pracují a byli navrženy na základě procesů, které se dějí kolem nás.

Genetické algoritmy slouží pro řešení optimalizačních problémů. Vychází z principů genetiky a mechanismech přirozeného výběru. Na rozdíl od matematických optimalizačních metod jsou genetické algoritmy velmi jednoduché. [50]

Genetické algoritmy jsou popsány pomocí chromozómu, genu, populace a fitness hodnotou, které nesou informace a hodnotí jejich kvalitu. Chromozóm je řetězec informací, který v sobě nese vlastnosti a chování každého jedince. Většinou jde o řetězec jedniček a nul, které mají zakódované pozice jedince v prostoru možných řešení. Nemusí se se vždy jednat o řetězec, ale mohou to být celá čísla, matice, vektory i křivky. [50]

Gen je nejmenší část chromozómu a při použitý v algoritmech je nedělitelný. Populace je skupina jedinců popsána chromozómy v rámci jedné generace. [50]

Fitness vyjadřuje číselnou hodnotu kvality každého jedince. V obvykle se jedná o číslo v intervalech od nuly do jedné, ale může to být číslo z libovolného intervalu. Pro každý problém musí být sestavena tzv. fitness funkce, jejíž výsledkem je požadována číselná hodnota. [50]

Pro genetické algoritmy se používají tři základní operace, které jsou selekce, křížení a mutace. Tyto operace se v dané generaci použijí nad celou populací a výsledkem je vznik nové generace. Tento proces se opakuje tak dlouho dokud se v nově vytvořené generaci nevyskytne jeden nebo více jedinců s požadovanými vlastnostmi. [50]

2.12 Fuzzy logika

Fuzzy logika byla nejprve definována jako fuzzy množina. Patří mezi jeden z používaných konceptů umělé inteligence. Pod slovem fuzzy si můžeme představit neostrý, matný, neurčitý. Cílem Fuzzy logiky je popsat realitu v její nepřesnosti a neurčitosti. [51]

Fuzzy logika umožňuje aproximovat lidské uvažování ve znalostních systémech. Reprezentace znalostí (logika prvního stupně, pravděpodobnostní teorie) klasickým přístupem není schopná zpracovat informace okolního světa, které jsou nepřesné a někdy špatně kategorizované. Tyto fakta vedli k vývoji fuzzy logiky, jejichž základními vlastnostmi jsou: [51]

- Přesné usuzování je speciální případ přibližného usuzování
- Vše je otázka míry
- Inference je možno vidět jako šíření pružných omezení
- Jakýkoliv logický systém může být fuzzikován

Dvě hlavní vlastnosti fuzzy systému, které zvyšují výkon ve specifických úlohách:

- Fuzzy systémy jsou vhodné pro nejisté nebo aproximační uvažování, především pro úlohy založené na matematickém modelu, který je příliš složitý na určení
- Fuzzy logika umožňuje rozhodování nad nekompletními nebo nejasnými instrukcemi

Při zpracování nepřesných nebo špatně definovaných systémů, fuzzy logika využívá kvantifikované nebo stupňované výroky místo výroků, které jsou přesně definované (true or false). Fuzzy množina je množina, která každému výroku přiřadí specifikovaný stupeň pravdivosti; tvrzení jsou částečně pravdivá, částečně nepravdivá, tuto skutečnost následně definuje interval od nuly do jedné. Tento princip hodnocení umožňuje vyšší flexibilitu a přesnost. [35]

2.13 Teorie chaosu

Jedná se o matematickou vědní disciplínu, která se zabývá chováním nelineárních dynamických systémů, které jsou citlivější na počáteční podmínky. Tento jev se vyskytuje u jevu „motýlí efekt“, který vychází z předpovědi počasí Edwarda Lorenza. Model systému je deterministický tzn. že je dobře definovaný a neobsahuje žádné náhodné parametry. [34]

Příklady takových systémů zahrnují atmosféru, solární systém, turbulenci tekutin, vývoj populace atd. [34]

Systémy, které prokazují deterministický chaos, jsou v jistém smyslu složitě uspořádané. Příbuzným oborem je kvantový chaos, který studuje chaotické chování v kvantových systémech. [34]

3 KLASIFIKACE DAT

Klasifikace tvoří rozsáhlou oblast analýzy dat a umožňuje nám určit do které skupiny (třídy, množiny) patří dané subjekty či objekty. Klasifikace je využívána v mnoha oblastech jako je např. medicína (slouží k odhalení demence na základě kognitivních testů) a v mnoha dalších oblastech (rozpoznávání osob podle otisku prstu nebo obličeje).

Klasifikaci dat zpravidla předchází roztřídění (nachystání) dat, které zahrnují vypořádání se s chybějícími hodnotami či odlehlými hodnotami, zároveň i transformaci dat a případně i další úpravy dat. Dále následuje redukce dat, která umožní vyjádření proměnných pomocí menšího počtu skrytých proměnných(extrakce) nebo pomocí proměnných z původního souboru (selekce). [29,30]

Cílem klasifikace dat je: [30]

- rozhodnutí o typu či charakteru objektu
- posouzení kvality stavu sledovaného objektu
- rozhodnutí o budoucnosti objektu

Termíny klasifikace a predikce jsou v jednotlivých vědních oborech chápány jinak a často jsou zaměňovány. Predikce nese charakter předpovědi či prognózy, předpovídá co se stane v budoucnosti. Pojem klasifikace používáme v případě, pokud vybíráme identifikátor klasifikační třídy z určitého konečného počtu možných identifikátorů. [29,30]

3.1 Rozhodovací stromy

Rozhodovací stromy nebo-li Decision Tree je graficko-analytická metoda. Slouží pro binární kvalifikaci objektů na základě jejich vlastností Jsou jednoduché na pochopení a interpretaci výsledků díky grafickému zobrazení, které pomáhá uživatelovi pochopit pozorovaný problém. [37]

Rozhodovací stromy hodnotí objekt pomocí souboru otázek, kde každá další otázka závisí na odpovědi aktuální otázky. Tento způsob otázek je vhodný pro nemetrické data, protože všechny otázky můžeme pokládat typu ano/ ne nebo pravda/ nepravda, a nebo také pomocí vlastností charakterizující sadu hodnot, které nevyžadují číselné hodnoty. Tento soubor otázek se zobrazuje v řízeném nebo v jednoduchém rozhodovacím stromě. [37,52]

Zobrazení rozhodovacího stromu je tvořeno pomocí orientovaného grafu, který se skládá z uzlů a hran. Každý uzel ve stromě znázorňuje klasifikační funkci v různých případech. Hrana ve stromě je hodnota, kterou může uzel očekávat. Stromy jsou zobrazované od vrcholu, kde se nachází kořenový uzel, který je spojený větvemi. Větve jsou navzájem spojené s dalšími větvemi anebo s koncovým uzlem. [37,52]

Klasifikace určitého objektu(vzoru) začíná v kořenovém uzlu, který se ptá na konkrétní vlastnost objektu a z toho to uzlu vede konečný počet hran. V každém následujícím kroku je objekt otestovaný podle otázky v aktuálním uzlu rozhodovacího stromu a pokračuje po uzlu, který je shodný s konkrétním výsledkem otázky. Na základě odpovědi pokračuje strom po následujícím nebo nadřazeném uzlu. Jakmile dojdeme ke koncovému uzlu, tak je kvalifikovaný třídou pro daný koncový uzel rozhodovacího stromu. [37,52]

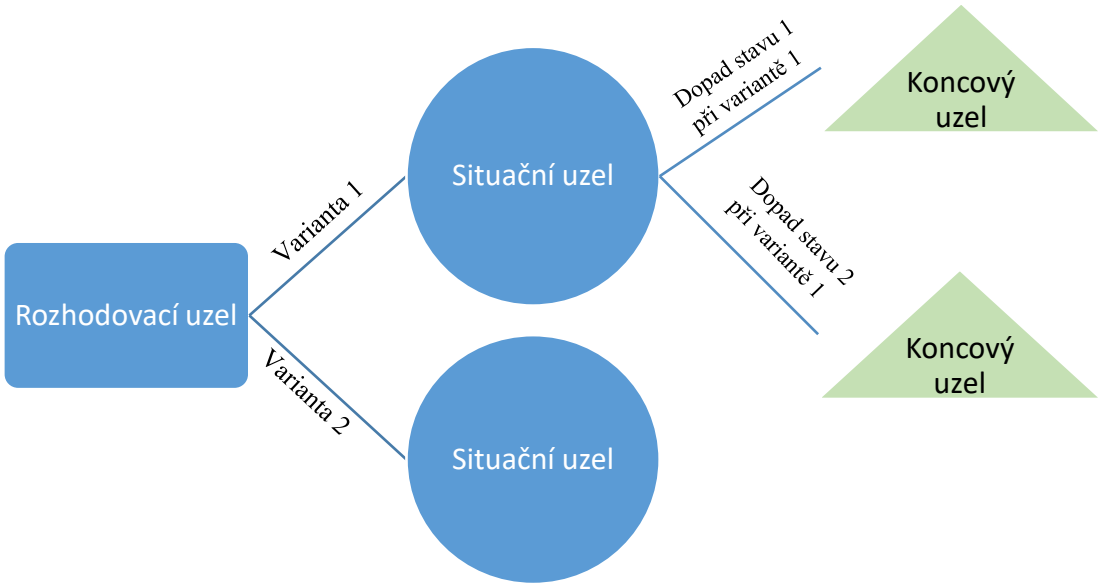
Uzly se dělí do tří skupin, které jsou: [37,52]

- Rozhodovací uzel je fáze rozhodování, při které se subjekt rozhodování rozhodne sám pro nějakou možnou variantu řešení, je zobrazen pomocí čtverce.
- Situační uzel je stav, kdy rozhodování závisí na působení náhodných faktorů, typické je pro zobrazení je kruh
- Koncový uzel je konečný stav řešení, který je zobrazen pomocí trojúhelníku

Na základě vstupní a výstupní proměnné je možné je dělit na kvalifikační a regresivní stromy. Oba dva typy mají určité rozdíly při určení místa, kde strom rozdělíme. [37,52]

Kvalifikační stromy dělí proměnné do dvou hlavních skupin, které můžeme číselně vyjádřit 0 nebo 1, ale i také Ano a Ne. Podle způsobu vyjádření dělíme výsledek na kategorický a diskrétní. Tento typ používáme v případě, když máme klasifikační problém. [37,52]

Regresivní stromy se používají, když je proměnná spojitá nebo číselná a ne kategorická. Příkladem je odhad ceny domu, jedná se o spojitou proměnnou, jejíž výsledek záleží na dalších spojitých proměnných, např. rozloha domu, lokalita. Tento typ stromu používáme, když máme problém s odhadem (predikce). [37,52]



Obr. 8) Struktura rozhodovacího stromu [52]

Existuje několik kritérií pro rozhodovací kvalifikační stromy, které jsou charakterizovány atributy. Tyto kritéria se používají v algoritmech pro rozhodovací stromy. Pro správné rozhodnutí o vhodném atributu pro větvení se používají tyto kritéria: Entropie, Informační zisk (Information gain), Gini index. [37,52]

Entropie je definována jako míra neuspořádanosti zkoumaného systému nebo také neurčitostí daného procesu. Entropii vyjadřuje vztah:

$$H = -\sum_{c=1}^{J} p_i \log_2 p_i \tag{2}$$

Kde p_i je relativní početnost třídy c na určité množině a J je počet tříd. Pro většinu stromů se vybírá atribut s nejmenší entropií.

Informační zisk vyjadřuje rozdíl entropie pro cílový atribut (pro celá data) a pro atribut o kterém se rozhoduje. Informační zisk měří redukci entropie způsobenou volbou atributu A.

$$IG(T, a) = H(T) - H(T|\mathrm{a}) \tag{3}$$

Kde *H(T|a)* je podmínka entropie vzhledem na hodnotu proměnné *a*.

Gini index měří míru nebo pravděpodobnost nesprávné klasifikace konkrétní proměnné, jestliže je daná proměnná vybraná. Toto kritérium se používá nejčastěji pro kvalifikační stromy typu CART. Pokud víme, že index Gini je 0, tak na základě toho to poznatku dokážeme vydedukovat, že všechny prvky patří do dané konkrétní třídy. Tento jev se nazývá, že třída je čistá. Naopak index Gini roven1 znamená, že prvky jsou náhodně rozdělené do různých tříd.

$$Gini(N) = \sum_{i \neq j} P(w_i)P(w_j) = 1 - \sum_{j} P^2(w_j) \tag{4}$$

Kde P vyjadřuje pravděpodobnost, že objekt je přiřazený do konkrétní třídy. [37,52]

3.1.1 Algoritmus ID3, C4.5 a C5.0

Algoritmus ID3 je základním algoritmem rozhodovacích stromů. Kritériem pro větvení je informační zisk. Lagoritmus ID3 byl rozšířen na algoritmus C4.5, aby bylo možné pracovat s numerickými atributy, chybějícími hodnotami a brát v potaz cenu za chybná rozhodnutí. [37,52]

Dalším rozšířením vznikl algoritmus C5.0 také označován See5, který je vhodný pro všechny typy proměnných, na výstupu vykazuje pouze kategoriální hodnoty. C5.0 je vhodné využít v případě velkého množství vstupních polí. Je schopen rychle odhadnout klasifikační model. Nabízí metodu (boost), která zvyšuje přesnost kvalifikace. Kritériem pro větvení je informační zisk a entropie [37,52]

3.1.2 Algoritmus CART

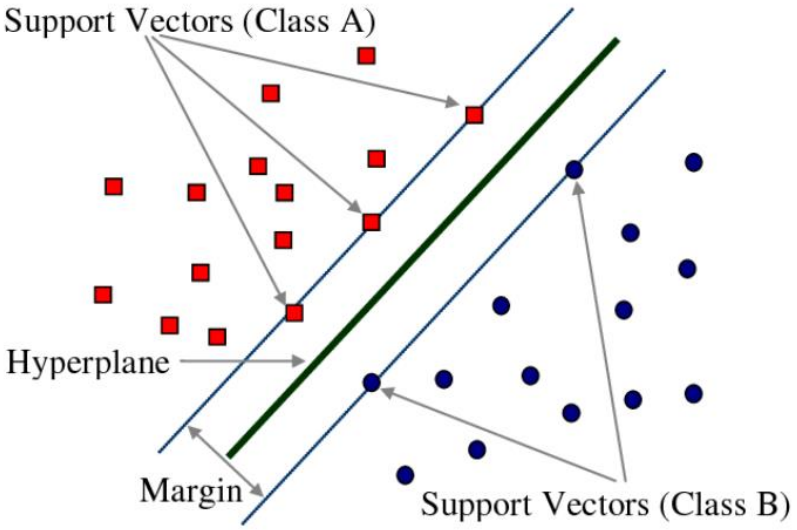
Stromy tohoto typu jsou vhodné pro regresivní, tak i pro klasifikační úlohy. Rostou na základě binárního dělení. Data vychází z jednoho kořenového uzlu a dále se dělí do dvou dceřiných uzlů na základě určitého dělícího kritéria. Pokud jsou uzly terminální, tak data v těchto uzlech konečná homogenní nebo se opět binárně dělí do dalšího kritéria.

Cart dokáže pracovat s diskrétními, tak spojitými výstupy. [37]

3.2 Support vector machines

Klasifikační metoda známá také ve zkratce SVM nebo-li metoda podpůrných vektorů je jedna z nejpopulárnějších metod v oboru strojového učení. Jedná se o algoritmus využívající principy učení s učitelem, proto je nutné mít množinu dat s jejími popisy na natrénování. [36]

Metoda je založena na nalezení hyperplochy v N-rozměrném prostoru (ve 2D prostoru se jedná o přímku), která odděluje dvě rozdílné třídy a neobsahuje žádné prvky. [36]



Obr. 9) Rozdělení vzorků do dvou tříd pomocí hyperrovin [53]

Na obr.8 jsou zobrazené dvě hyperroviny, kde jedna je s úzkým okrajem a druhá se širokým. Předpokládáme, že testovací vzorek má být zařazen do třídy A. Hyperrovina 1 zařadila vzorek správně do třídy A a hyperrovina 2 ne. Rozdělení dat do dvou tříd je jednoduché, pokud jsou data lineárně separované (reálná data nejsou většinou taková). [36]

Zavedení tohoto algoritmy ve 2D prostoru, kde hranice mezi třídami je tvořena přímkou. Tuto přímku můžeme vyjádřit následující rovnicí.

$$y = \omega \cdot x + b, \tag{5}$$

Kde ω je váhový vektor, x je vstupní vektor a b je hodnota zvaná bias, která posouvá danou přímku.

Body nacházející nad touto přímkou můžeme označit jako + a rovnicí je lze vyjádřit jako

$$\omega \cdot x + b > 0 \tag{6}$$

Body nacházející se pod přímkou lze označit jako – a rovnice která je vyjadřuje je

$$\omega \cdot x + b < 0 \tag{7}$$

Algoritmus nehledá pouze body nad a pod přímkou, ale do výpočtu je nutné zahrnout body z již zmíněného okraje M. Obě rovnice musí být upravené do tvaru

$$\omega \cdot x + b \geq +M \tag{8}$$

$$\omega \cdot x + b \leq -M \tag{9}$$

Jestliže najdeme bod z množiny x^+, který se nachází nejblíže k hledané rozdělovací přímce, tak získáme tzv. support vector. Support vector z množiny x^- je vzdálený od dělící přímky ve vzdálenosti M: Vzdálenost mezi support vectory z množin bodu x^+ a x^- je přesně $2M$. Na základě těchto poznatků můžeme konstatovat, že ω je kolmý k separační přímce a k linii okrajů hranic + a -. Tento vztah popisuje následující rovnice

$$x^- = x^+ + v \cdot \omega \tag{10}$$

Další rovnici lze odvodit ze vztahu, že $x^+ + x^- = 2M$, která popisuje rovnici kvalifikační přímky

$$M = \frac{1}{2|\omega|} = \frac{1}{2\sqrt{\omega \cdot \omega}} \tag{11}$$

Pomocí rovnice kvalifikační přímky můžeme spočítat velikost okraje *M*. Hledáme takový vektor přímky ω a velikost *b,* aby velikost okraje M byla co největší. Z rovnice plyne, že největší velikost *M* je, když $\omega \cdot \omega$ bude co nejmenší. Jestli je toto jediná podmínka, tak je nejjednodušší dosadit za $\omega=0$, tak by byl problém vyřešen. Ale není to jediná podmínka, další důležitou podmínkou je, aby ω rozdělovala množinu bodů + a -. Nastává další rozhodovací problém nalézt co největší velikost okraje *M* a nejnižší vektor ω, aby separační linie rozdělovala množinu bodů správně. [36]

3.3 K- nearest neighbour

Metoda nejbližších sousedů je nejjednodušší metoda strojového učení, která je řazena mezi neparametrické metody klasifikace, která využívá učení s učitelem. Při použití metody není nutné znát rozložení pravděpodobnosti testovaných dat a chyb, které vznikají během použití metody. Chyby jsou srovnatelné s chybami složitějších metod. Daná metoda se využívá v mnoha oborech např. v molekulární biologii. [16]

Nejprve je nutné spočítat si matici míry nepodobnosti mezi prvky- matice D. Jedná se o matici typu *n x n*, která má na místě *i, j* prvek $d(X_i,X_j)$, kde *d* je námi vybraná míra nepodobnosti. Díky symetrii matice a nulovosti prvků na hlavní diagonále, stačí uvažovat jen prvky po hlavní diagonálou. Prvky seřadíme podle velikosti do neklesající posloupnosti

$$d_{(1)} \leq d_{(2)} \leq \cdots \cdots \leq d_{\binom{n}{2}} \tag{12}$$

První rozklad $S^{(0)}$ je rozložen na jednoprvkové shluky.

$$S^{(0)} = \{X_1, X_2, \cdots \cdots X_n\} \tag{13}$$

Dané kroky se neustále opakují do doby, dokud nedosáhneme jediného shluku prvků.

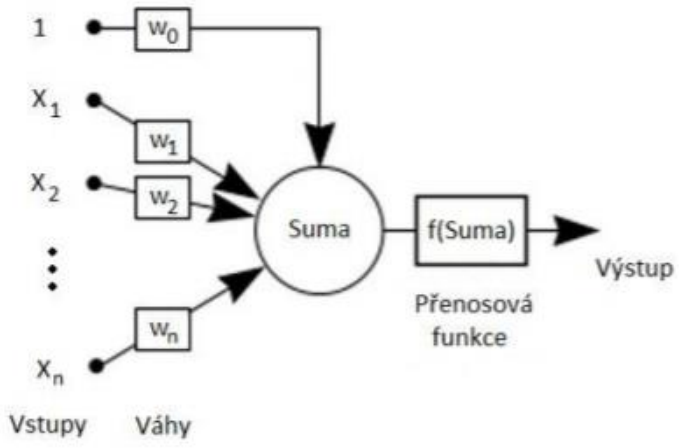
V k-tém kroku $(1 \leq k \leq \binom{n}{2})$ použijeme prvek posloupnosti $d_{(k)} = d(X_i, X_j)$. Pokud objekty X_i a X_j patří do stejného shluku následuje krok *k+1*. Jestliže objekty X_i a X_j patří do dvou různých shluků, tak tyto shluky spojíme, jelikož si jsou vzájemně nejblíže (ve smyslu nejbližšího souseda).

Pro hledání nejbližšího souseda v množině je možné použít několik druhů metrik, nejpoužívanějšími je Euklidova metrika nebo Hammingova metrika. [38]

3.4 Klasifikace pomocí neuronové sítě

Neuronové sítě jsou úzce spjaty s umělou inteligencí, představují algoritmy inspirované strukturou lidské nervové soustavy. Vychází ze základního modelu neuronu, které jsou uspořádány do libovolného počtu vrstev a každá vrstva odpovídá konkrétním požadavkům. Vrstvy nacházející se mezi vstupní a výstupní částí slouží k většině k výpočtů, ale jsou považovány za skryté. [22]

Neuronová síť dokáže přijímat několik datových vstupů současně, na kterých je poté prováděn proces trénování, který umožní vypozorovat z dat charakteristické vzorce chování. Tyto vzorce nám umožňují v budoucnu předpovědět výstup dat pro netrénovanou datovou sadu.[22]



Obr. 10) Umělý neuron [4]

Klasifikace pomocí neuronové sítě je poměrně jednoduchý proces. Neuronové síti je poskytnut feature vektor na vstupu a kategorizace se objeví na výstupu. Každý výstup má přidělenou klasifikační třídu. Jednotlivé výstupy hodnot popisují s jakou pravděpodobností je si síť jistá svým rozhodnutím. Pokud výstupní signál nedosahuje cílové hodnoty, tak klasifikace může být odmítnuta, aby bylo dosáhnuto vyšší spolehlivosti výsledku. Výstup neuronu je definovaná vztahem:

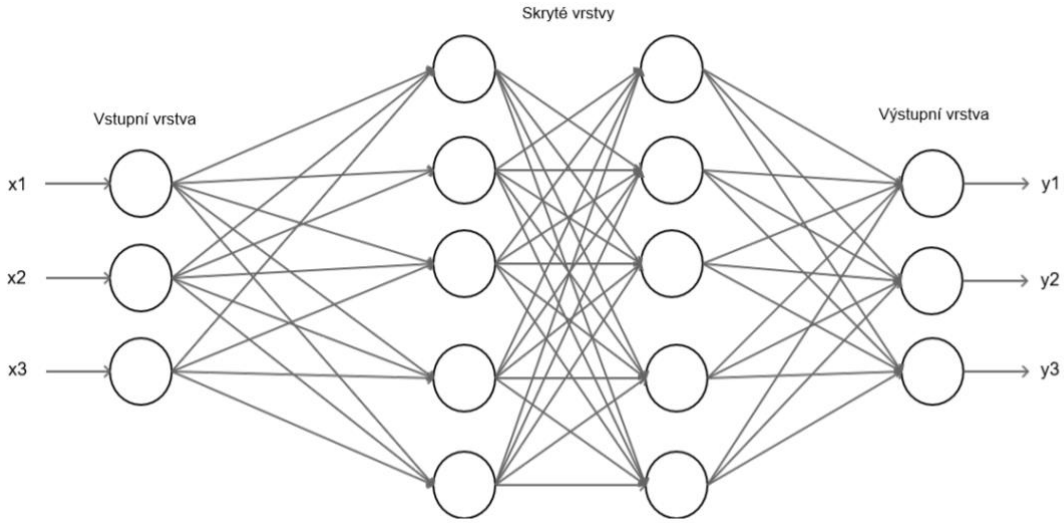
$$0 = f(net) = f(\sum_{j=1}^{n} w_j \cdot x_j), \tag{14}$$

kde w_j je váhový vektor, funkce $f\ (net)$ odkazuje na přenosovou funkci, x_j jsou vstupy neuronové sítě a 0 je výstupem sítě. [4]

Hlavním problém při implementaci neuronových sítí je samotný návrh sítě. Lze říci, že je možné neuronovou síť jakkoliv komplexně zkonstruovat, ale nelze předpovědět, zda bude daná síť vynikat v klasifikačním problému. [33]

Existuje několik typů neuronových sítí, jako je Perceptron, Hopfieldova síť, Kohonenova síť , ART síť a další. V této práci bude použita vícevrstvá neuronová síť neboli percepton. [22]

Percepton je nejjednodušší formou neuronové sítě. Jedná se o vícevrstvé sítě, které se skládají z vrstev neuronů. Vrstvy jsou mezi sebou vzájemně propojeny tak, že neuron z jedné vrstvy je propojen se všemi neurony následující vrstvy. V rámci jedné vrstvy nejsou definována žádná propojení. [5]



Obr. 11) Vícevrstvá neuronová síť [33]

Počet neuronů vstupní vrstvy odpovídá velikosti vstupu a analogicky počet neuronů výstupní vrstvy je postaven na kódování výstupu. Mezi vstupní a výstupní vrstvou se v tomhle případě nachází dvě skryté vrstvy neuronů. Jestliže je zvolen nízký počet neuronu ve skrytých vrstvách, tak dochází k tomu, že síť není schopná „naučit“ se daný problém a poté správně klasifikovat. V opačném případě, pokud je zvolen vysoký počet neuronů, tak může dojít přeučení sítě a model nebude fungovat správně. [5,22]

4 SW PROSTŘEDÍ

4.1 Matlab

Matlab (MATrix LABoratory) je interaktivní prostředí a skriptovací jazyk čtvrté generace, který je vhodný pro vědecké a inženýrské výpočty, modely, simulace, vývoj algoritmů a vizualizaci dat. [42]

V softwaru Matlab nalezneme široké spektrum klasifikačních algoritmů jako jsou rozhodovací stromy, algoritmy pro lineární a kvadratickou diskriminační analýzu (LDA, QDA), algoritmy podpůrných vektorů (SVM), naivní bayesovský klasifikátor a algoritmus k nejbližším sousedům. Je možné využít i obecnější přístup, jako např. neuronové sítě, fuzzy logiku atd. [41, 42]

V prostředí Matlab je práce s klasifikátory snadná, i když se jedná o odlišné algoritmy, Matlab nad nimi poskytuje jednotné rozhraní. Nejprve si zvolíme typ klasifikátoru. Klasifikátor vytvoří a naučí funkci fit doplněnou o jméno klasifikátoru (např. fittree pro klasifikační strom). Model je vytvořen jako objekt v pracovním prostředí Matlabu. Pro následnou klasifikaci nových dat slouží funkce predict, aniž by záleželo na typu klasifikátoru. Je možnost nastavení různých parametrů, které ovlivňují klasifikační algoritmy.[41]

V Matlabu je k dispozici přehledné grafické uživatelské rozhraní, Classification Learner App. Tento program umožňuje vybírat data, volit a nastavovat klasifikátory a realizovat vlastní proces naučení se modelu. Dále je umožňuje přehledné zobrazení výsledků včetně metrik pro porovnání jednotlivých klasifikátorů mezi sebou. [41]

4.2 Python

Python je moderní programovací jazyk, jehož autorem je Guido van Rossum. Je vyvíjen jako open source projekt, tzn. že nabízí zdarma instalační baličky pro většinu běžných platforem (Linux, MS Windows, Android, macOS). [40]

Jedná se o velmi populární programovací jazyk díky jeho čistému a čitelnému kódovacímu stylu. Jednoduché čtení umožňuje především, že kód musí být odsazen pro definování funkcí, smyček a řídících struktur. [40]

Hlavními výhodami použití Pythonu jsou: [40]

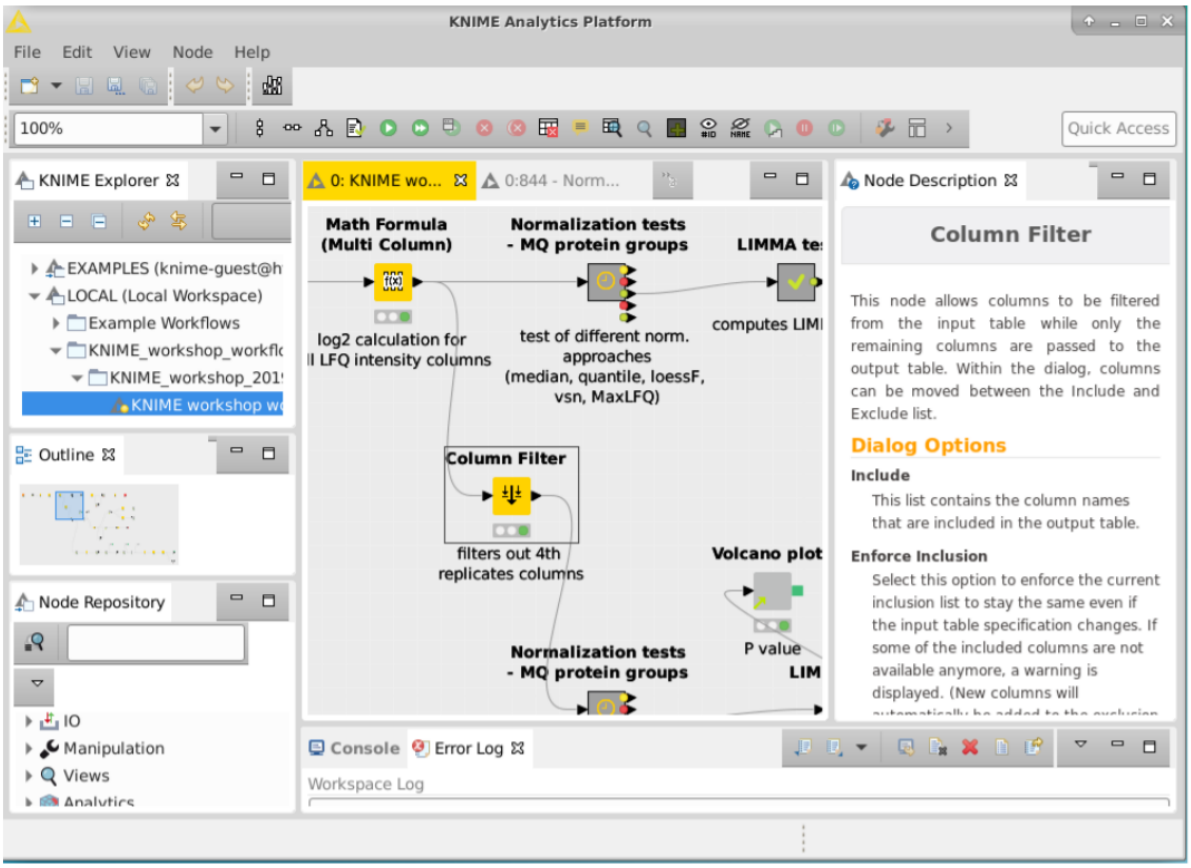
- Není nutné dopředu deklarovat typ proměnných
- Jednoduchá a čitelná syntaxe
- Obsahuje několik knihoven pro načítání a vizualizaci dat, statistiku a další
- Je využíván širokou skupinou programátorů, dobrá dokumentace

4.3 Knime

Jedná se o volně přístupnou analytickou platformu pro zpracování dat. Pomocí vizuálního programování je možné vytvořit celý pracovní proces od nahrání souborů s daty, přes předzpracování dat a statistické testy, až po vizualizaci a vytvoření potřebného výstupu.[43]

Knime nabízí pracovní jednotky nebo-li nody a komponenty umožňující import a export souborů, manipulaci s tabulkami (např. filtrování, přejmenování, řazení, přidáváním sloupců, spojování tabulek), náhled grafů i skriptovací nody a další. Jednotky je možné mezi vstupními

a výstupními porty mezi sebou propojovat, které mohou být v případě vlastních jednotek přidány, odebrány nebo změněn jejich datový typ (např. obrázek, proměnná či data v tabulce). [43]



Obr. 12) Prostředí KNIME [43]

Na obr.12 je vyobrazeno prostředí KNIME s otevřeným workflow s propojenými pracovními jednotkami ve středu. Vlevo dole je repositář nodů, náhled workflow, prohlížeč složek. Nahoře je ovládací lišta a vpravo je popis vybrané jednotky. [43]

4.4 GNU Octave

Software Octave slouží pro numerické řešení problémů lineární algebry, hledání řešení nelineárních rovnic, integrování funkcí a práce s polynomy pod veřejnou licencí GNU General Public Licence (GPL). GNU Octave je do velké míry podobný programu Matlab. Největší rozdílem je především to, že Octave podporuje autoinkrementaci stejně, tak jako jazyk C. Matlab bohužel tuto podporu nemá. Dalšími rozdíly jsou odlišné syntaktické zápisy, případné volání jinak pojmenovaných funcích. [39]

Hlavní důvodem, proč byl zvolen software GNU Octave pro praktickou část je, že je veřejný a je zdarma. Mnoho uživatelů neustále vyvíjí další rozšiřují balíčky pro potřeby matematiky, statistiky, simulací a dalších jiných vědeckých odvětví. [39]

5 ZPRACOVÁNÍ DAT

Hlavním cílem diplomové práce je implementace klasifikačních metod na vygenerovaný neroztříděný soubor dat. Zpracovávání dat je prováděno v programu GNU Octave z licenčních důvodů. Jelikož GNU Octave je dostupný pro kohokoliv, tak je možné v budoucnu zpracované skripty použít na různý soubor dat v jakékoliv firmě.

Existuje široké spektrum klasifikačních algoritmů. Pro klasifikaci simulovaných dat jsem implementovala celkově tři klasifikační algoritmy, jmenovitě: algoritmus podpůrných vektorů (SVM), algoritmus k nejbližším sousedovi (KNN) a klasifikace pomocí neuronové sítě.

Všechny tři zvolené klasifikační metody byly podrobně popsány v teoretické části.

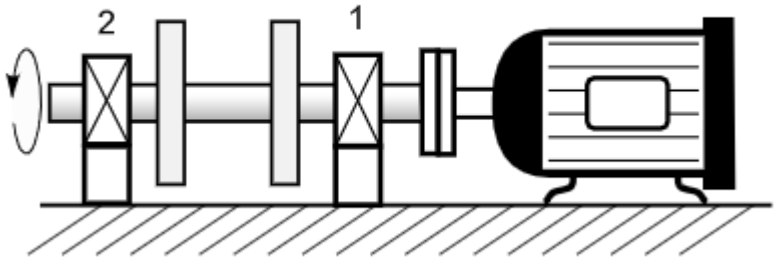
5.1 Vstupní data

Vstupními daty pro praktickou část jsou dva typy souborů dat. Jedná se o:

- Data simulovaná pro otestování metodiky
- Data získaná z reálného měření

Z každého typu poskytnutých dat byla náhodně vybrána polovina záznamů pro učení a druhá pro otestování, abychom mohli ověřit správnost implementovaných klasifikačních modelů.

Simulovaná data byla vygenerována na základě simulace konkrétních poruch na rotačních strojích v simulačních softwaru. Reálná data jsou hodnoty, které vznikly měřením odchylek polohy rotačního stroje v ose X a Y při dynamické a statické nevývaze. Data jsou získané ze dvou analogových akcelerometrů umístěných na stojanech ložisek.

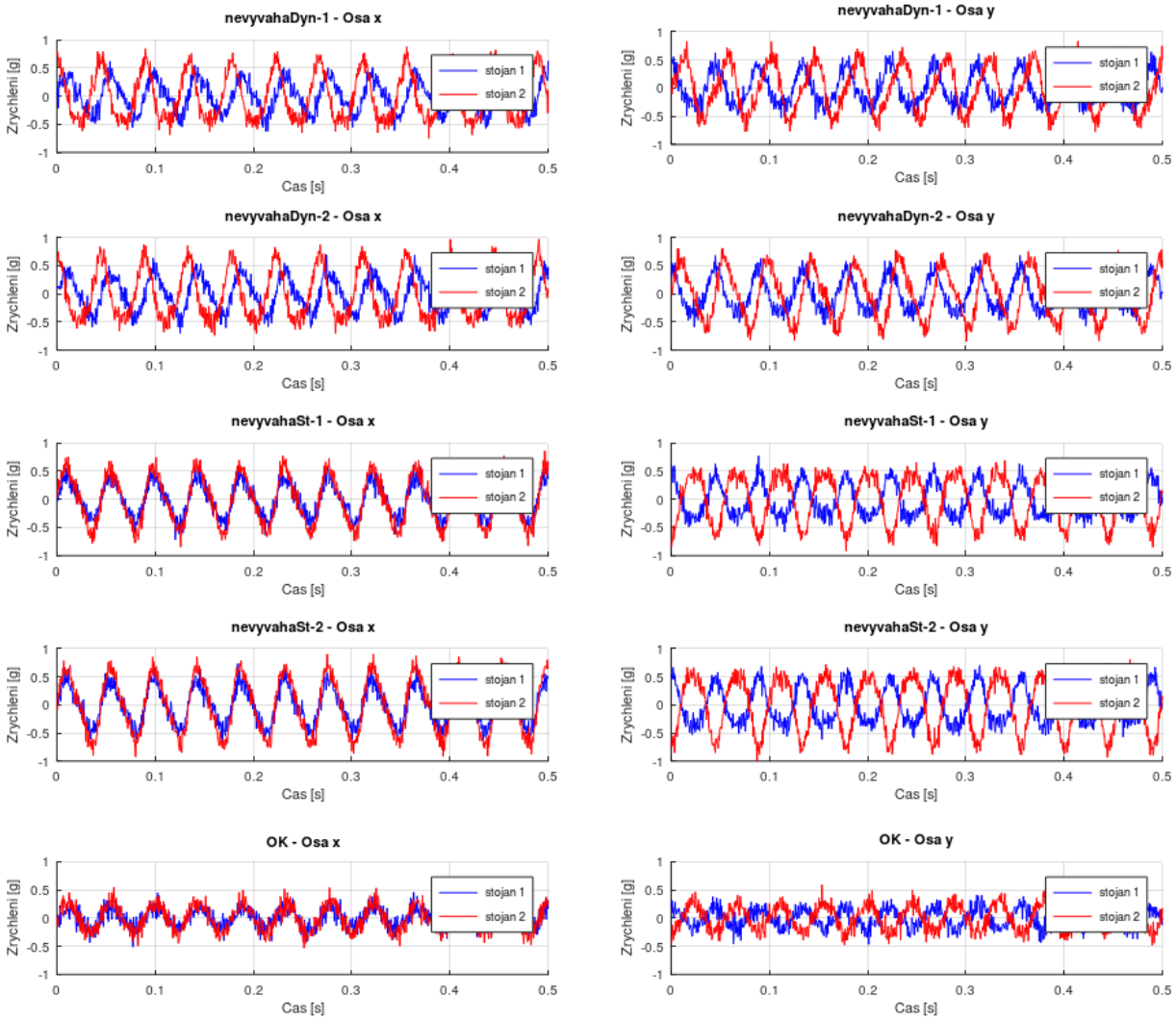


Obr. 13) Umístění akcelometrů poz.1 a poz.2 [45]

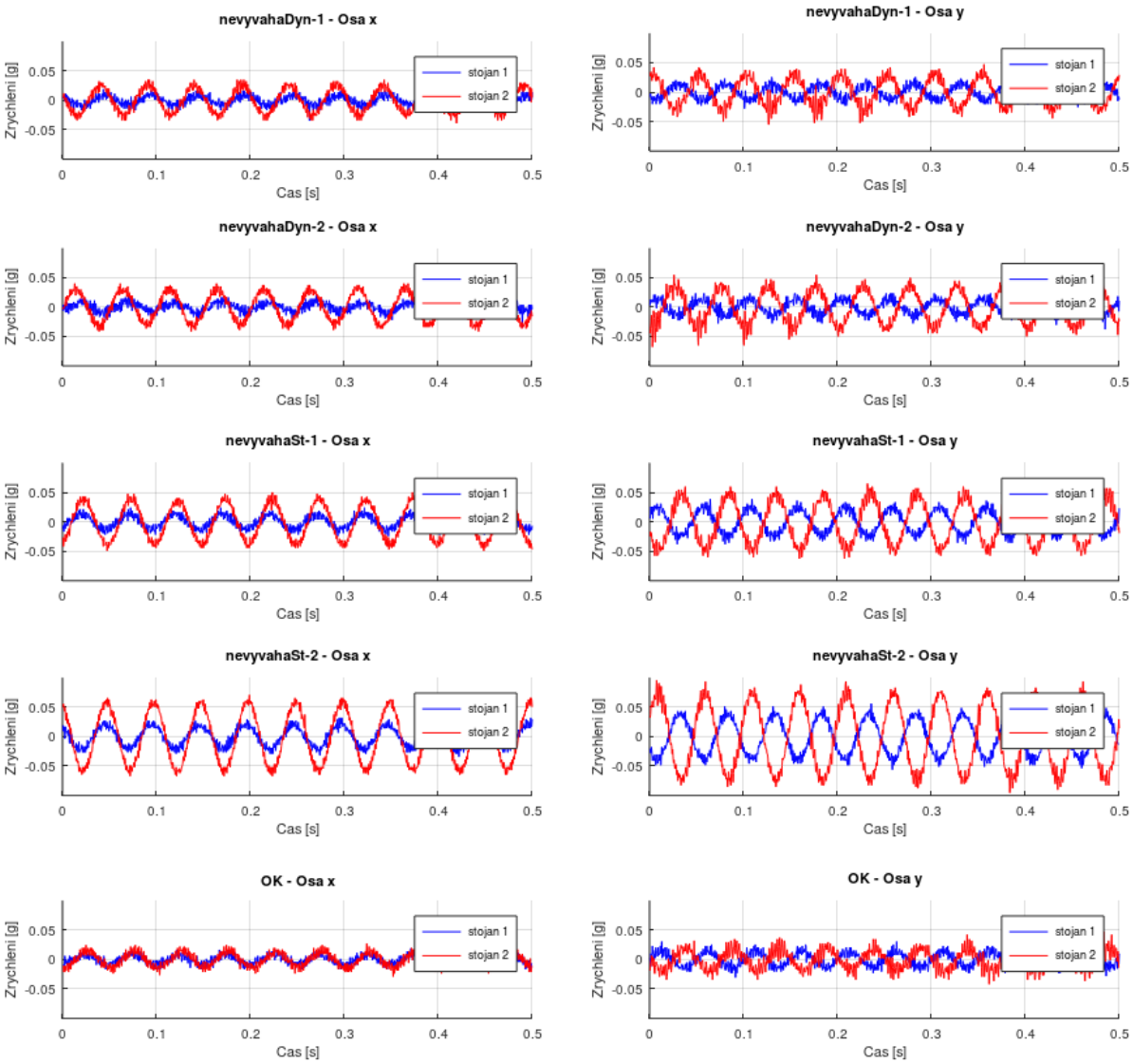
Hodnoty polohy byly snímány vzorkovací frekvencí 2000 Hz s nastavením 1000 vzorků. V každém souboru dat jeden řádek představuje jedno měření. V každém datasetu (pro každou osu na každém stojanu) najdeme 1000 měření o 1000 hodnotách. Celkově tedy 40 000 měření. [45]

Všechna vstupní data byla získána ze dvou stojanů ložisek (1,2) v ose X (1x, 2x) a Y(1y,2y), kde byli ve dvou úrovních nastaveny základní poruchy (statická (St1, St2) a dynamická (Dyn1, Dyn2) nevývaha), která jsou porovnávány s bezchybným (OK stav) stavem. Celkově tedy máme 5 tříd pro klasifikaci jak pro testovací, tak i naměřená data.

Obr. 14) Znázornění vstupních dat

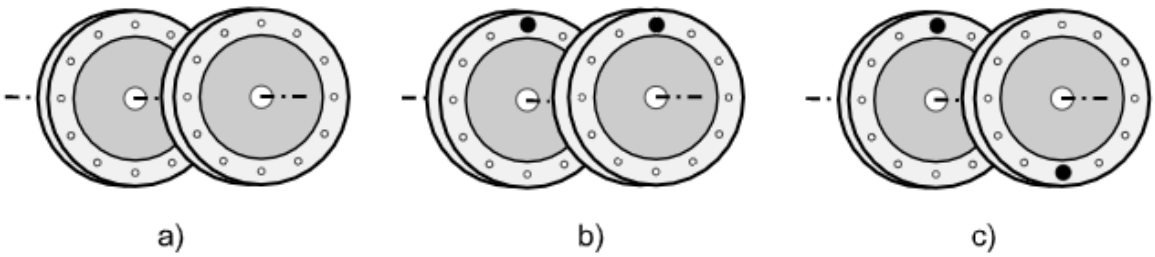


Obr. 15) Průběh všech stavů náhodného vzorku ze simulovaných dat



Obr. 16) Průběh všech stavů náhodného vzorku z naměřených dat

Statická a dynamická nevývaha se od sebe liší především v posunuté fázi signálu mezi stojany ložisek, které jsou ve stejné ose. Rozdílem při simulaci dynamické a statické nevývahy je umístění závaží na rotační disky. Závaží mají rozdílnou hmotnost a rychlost otáčení u všech měření byl nastavena na 1190 ot/min. [45]

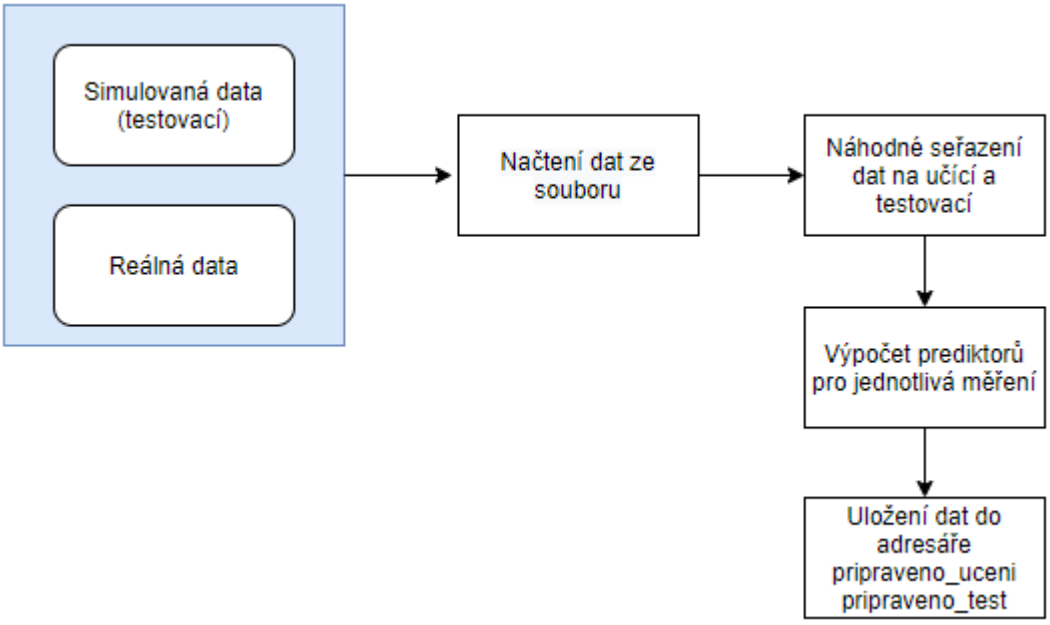


Obr. 17) Schéma závaží a jejich poloh a) OK stav b) Statická nevývaha c) dynamická nevývaha [45]

5.2 Předzpracování dat

Pro samotnou klasifikaci je nutné data si předpřipravit (roztřídit). Cílem předzpracování dat je vybrat hodnoty (údaje), které jsou podstatné pro klasifikaci a prezentovat je ve tvaru vhodném pro zpracování klasifikačních modelů. Těmito reprezentujícími hodnotami jsou takzvané prediktory. Prediktory v podstatě shrnují každé z měření do několika charakteristických hodnot. Výběr prediktorů a správná implementace jejich výpočtu je zásadní krok pro úspěšnou klasifikaci jednotlivých měření klasifikátory.

Z každého typu poskytnutých dat byla náhodně vybraná polovina dat pro trénování a druhá pro otestování, abychom poté mohli ověřit správnost učícího modelu. Na obr. 18 je znázorněn postup průběhu předzpracování dat pro následnou klasifikaci.



Obr. 18) Algoritmus předzpracování dat

Nejprve byly implementovány prediktory, které slouží pro klasifikaci jednotlivých tříd. Jedná se o prediktory RMS (Root Mean Square) - používá se ve vibrodiagnostice pro stanovení stavu stroje, STDV (Standard Deviation and Variance) a FFT (Fast Fourier transform). Skript pro výpočet prediktorů je k nahlédnutí v přiloženém zdrojovém kódu.

```
function y = vypocet_prediktoru(mereni)
    T_rms  = Wrms(mereni);
    T_stdv = Wstd(mereni);
    ##    T_pca  = Wpca1(mereni);
    T_fft  = Wfft(mereni); % Frekvence s nejvysi amplitudou
    y = [T_rms, T_stdv, T_fft];
end
```

Obr. 19) Výpočet prediktorů

Prediktor FFT

Rychlá Fourierova transformace je efektivní metoda pro zpracování různých signálů. Jedná se o algoritmus pro zpracování diskrétní Fourierovy transformace (DFT) a její inverze. [46]

Diskrétní Fourierova transformace je diskrétní spektrum, které obsahuje konečný počet harmonických složek. Nevýhodou této transformace je časová náročnost, která roste se čtvercem délky vstupní posloupnosti. Kvůli časové náročnosti vznikla Rychlá Fourierova transformace, která vychází z vlastností exponenciálních funkcí a výrazně snižuje potřebnou dobu výpočtu. [46]

```
FFT_out=fft(vstup, L, 2);

P2 = abs(FFT_out/L);
P1 = P2(:,1:L/2+1);
P1(:,2:end-1) = 2*P1(:,2:end-1);

[Amax,Imax]=max(P1, [], 2);
Fmax= Fs*(Imax/L);
```

Obr. 20) Výsledná FFT

Výsledkem předzpracování jsou matice o velikosti 500x14 pro každou třídu měření, a to jak pro trénovací data, tak pro data testovací.

nevyvahaDyn-1	[500x14 double]
nevyvahaDyn-2	[500x14 double]
nevyvahaSt-1	[500x14 double]
nevyvahaSt-2	[500x14 double]
OK	[500x14 double]

Obr. 21) Výsledek předzpracování

Jednotlivé sloupce v maticích představují prediktory FFT, RMS, STDV pro každý stojan a v osách X a Y a fázový posun mezi stojany v obou osách, který představuje odhadované zpoždění mezi signály. Řádky reprezentují jednotlivá měření.

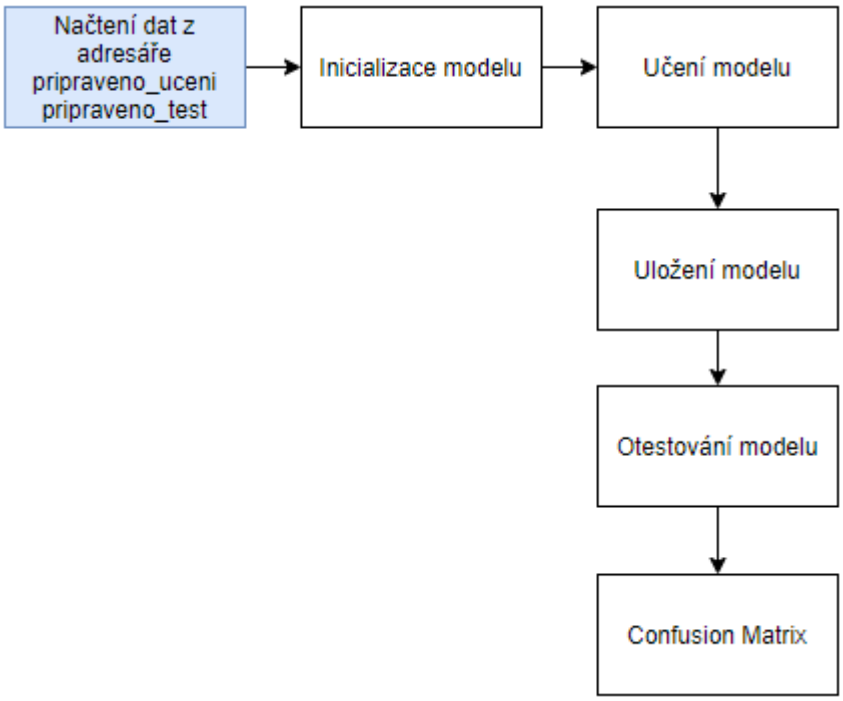
Celkově je tedy k dispozici 14 prediktorů (RMS_1X, STDV_1X, FFT_1X, RMS_2X, STDV_2X, FFT_2X, FAZE_X, RMS_1Y, STDV_1Y, FFT_1Y, RMS_2Y, STDV_2Y, FFT_2Y, FAZE_Y) pro 500 měření pro učící data a 500 měření pro data testovací.

	RMS_1X	STDV_1X	FFT_1X	RMS_2X	STDV_2X	FFT_2X	FAZE_X	RMS_1Y	STDV_1Y	FFT_1Y	
	1	2	3	4	5	6	7	8	9	10	....
1	0,28987	0,29001	1,2	0,43274	0,43295	1,2	-0,2	0,2997	0,29985	1,2	....
2	0,29623	0,29637	1,2	0,43141	0,43163	1,2	0,21	0,30055	0,3007	1,2	....
3	0,27099	0,27113	1,2	0,38573	0,38592	1,2	-0,19	0,2699	0,27004	1,2	....
4	....	....	....	....	....	....	....	....	....	....	....

Tab 1) Ukázka matice prediktorů pro jednotlivá měření

5.3 Implementace klasifikačních metod

Všechny modely, které byly zvoleny pro klasifikaci dat musí být nejprve natrénovány na učící množině dat a poté otestovány na množině testovací.



Obr. 22) Algoritmus klasifikačních modelů

Byly celkově vybrány a implementovány 3 klasifikační modely, které jsou K- Nearest Neighbours, SVM a mnou specifikovaná neuronová síť. Jednotlivé klasifikační modely jsou podrobně popsány v kapitole Klasifikace data a jsou založeny na principu učení s učitelem. Zejména u neuronových sítí musíme brát zřetel, aby nedošlo k přetrénování. Učení modelů se skládá ze 3 hlavních kroků, které jsou inicializace, učení a uložení.

```
knn_model = knn(20);
knn_model.train(ucici_data, ucici_tridy);
knn_export = knn_model.export();
```

Obr. 23) Hlavní kroky při učení klasifikačního modelu

První řádek představuje inicializaci modelu, tzn. že v tomto kroku volíme klasifikační metodu, kterou chceme použít. Zároveň jsou v tomto kroku definovány specifické parametry klasifikačního modelu. Výběr těchto parametrů hraje klíčovou roli při učení a následné klasifikaci každého modelu, proto je nutné vyladit parametry (často iterativním přístupem) co možná nejlépe.

Ve druhém řádku pomocí funkce train dochází k tomu, že se načtou předzpracovaná učící data, která jsou na základě naší znalosti (dáno vstupními daty) přiřazena k jednotlivým učícím třídám. Po naučení modelu takto poskytnutými daty poté pomocí „podobnosti“ dokáže model roztřídit úplně nová data, která v budoucnu poskytneme modelu bez přiřazené třídy.

Ve třetím kroku dochází k uložení výsledného modelu metody do souboru. Pro více informací týkajících se implementace jednotlivých kroků pro každý model lze nahlédnout do přiloženého zdrojového kódu.

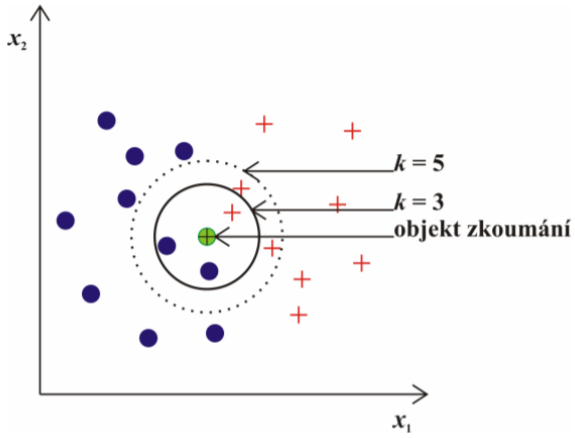
Pro vyhodnocení je využit další skript, ve kterém se každý z uložených modelů načte a otestuje se na testovacích datech. Výsledná klasifikace modelu je porovnána s námi známými třídami přiřazenými k jednotlivým měřením.

Vyhodnocení klasifikačních algoritmů lze v některých jednoduchých případech porovnávat pomocí srovnávacích vztahů, ale těchto případů je velmi málo. Rozhodla jsem se pro použití sofistikovaného nástroje, kterým je confusion matrix („matice záměn“). Z této matice lze objektivně hodnotit kvalitu použitých algoritmů. Více bude o výsledcích modelů pojednáno níže.

5.3.1 KNN

Metoda vychází z předpokladu, že zkoumaný prvek patří do stejné třídy, jako jeho k-nejbližších sousedů. Pro změření vzdálenosti mezi prvky lze použít několik metrik (euklidovská, korelační, kosinová…). [49]

Jestliže je například nastavena hodnota k=3, tak bude přiřazen do jiné třídy, než když bude k=5, proto je důležité, aby hodnota k byla určená správně. Důležitým faktorem je volit za k liché číslo, z důvodu jednoznačnosti klasifikace. [49]



Obr. 24) Klasifikace pomocí K-NN [49]

Počet k-nejbližších sousedů byl interaktivně zvolen na základě úspěšnosti klasifikace.

K-NN simulovaná data					
Počet vybraných K-NN	15	19	25	31	37
Úspěšnost (%)	89,68	**90,12**	89,64	89,48	89,4

Tab 2) Úspěšnost klasifikace K-NN

V Tab 2) je zobrazena úspěšnost klasifikace v závislosti na počtu nejbližších sousedů *k* pro daná data. Pro výslednou klasifikaci jsem zvolila parametr *k*=19 z důvodu největší úspěšnosti klasifikace na dané množině dat.

5.3.2 SVM

Klasifikace pracuje za pomoci SVM kernel funkcí, které porovnávají všechny podpůrné vektory (tedy vektory xi , u nichž byl natrénován parametr $\alpha_i \neq 0$) se zadaným vektorem, který má být zařazen do jedné ze tříd. Zařazení do vhodné třídy je prováděno pomocí funkce sign(.) z diskriminační funkce. [47]

$$g(x) = \sum_{i=1}^{N} \alpha_i \cdot k(v_i, x) - \varrho, \tag{15}$$

Kde α_i je i-tý parametr stanovující příslušnost k jedné ze dvou funkcí, v_i je i-tý podpůrný vektor, x je klasifikovaný vektor, k (-) je použitá kernel funkce, ϱ je práh, ω je rozhodnutí klasifikátoru a N je počet všech podpůrných vektorů. [48]

Kernel funkce je použita z knihovny libSVM. Jedná se o programovanou knihovnu v jazyce C++ a v oblasti klasifikace ji lze považovat za standard. Knihovna používá pro trénování algoritmus SMO s modifikacemi od Keerthi. Je tedy velmi přesná, co se týče výsledků. [48]

Knihovna obsahuje přesně definované funkce, které vstupní soubory s trénovacími/ testovacími daty nebo natrénovaným modelem načtou a hlavním úkolem je funkci zavolat se správným parametrem.

Při klasifikaci libSVM modelu byl počet podpůrných vektorů 1024. Délka vektoru ovlivňuje přesnost modelu, která je následně vyobrazena v confusion matrix.

5.3.3 Neuronová síť

Klasifikace pomocí neuronové sítě je poměrně jednoduchý proces. Neuronová síť se skládá ze tří vrstev: vstupní, skrytá a výstupní vrstva. Množství vstupů je závislé na počtu interakcí, výstupní vrstva se skládá z pěti výstupů podle počtu tříd (stavů měření). Ve skryté vrstvě je použita sigmoidální přenosová funkce, ve výstupní vrstvě softmax přenosová funkce. [48]

```
fprintf("Uceni NN...\n")
nn_model = nn(200);
nn_model.train(ucici_data, ucici_tridy);
nn_export = nn_model.export();
```

Obr. 25) Model NN

Neuronová síť je trénovaná pomocí funkce train. Pro trénování ve skryté vrstvě je důležité zjistit potřebný počet iterací, aby se model stihl dostatečně natrénovat. Počet iterací se volí individuálně a musíme dbát na to, aby nedošlo k přeučení.

NN - simulovaná data					
Počet iterací	100	150	200	250	270
Úspěšnost (%)	79,36	79,52	**81,3**	80,4	79,84

Tab 3) Úspěšnost klasifikace NN

V Tab 3) je závislost úspěšnosti klasifikace NN na počtu iterací zpětného šíření chyby při učení (backpropagation). Jako optimální počet iterací se jeví 200, jelikož dosahuje největší úspěšnosti.

```
Uceni NN...
Iteration     2 | Cost: 2.517134e+00
```

Obr. 26) Načítání modelu NN

Na obr. 26 je zobrazeno učení modelu NN. Neuronová síť se učí po 200 iterací, kdy při každé je zobrazena aktuální hodnota chyby. Tu se pomocí učení snažíme minimalizovat, ale zároveň zamezit přeučení. V takovém případě se model naučí příliš přesně na konkrétní učící data a s mírně odlišnými testovacími by měl při klasifikaci problém. Hodnota počtu iterací byla zvolena po manuálním testování.

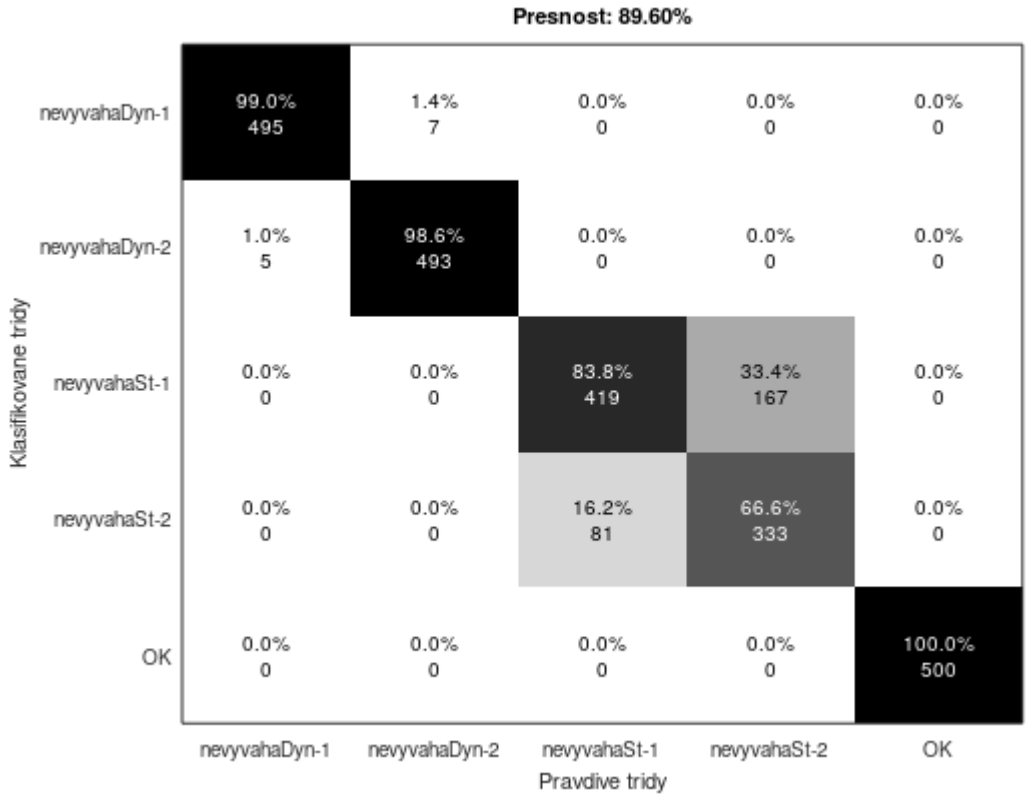
5.4 Zhodnocení klasifikačních metod

Porovnání jednotlivých klasifikačních metod je prováděno pomocí tzv. Confusion matrix. Jedná se o matici, která zobrazuje kompletní přehled klasifikace prvků do tříd. Matice je o velikosti A x B, kde A definuje počet pravdivých tříd a B je počet klasifikovaných (predikovaných) tříd. Matice znázorňuje kolik prvků bylo klasifikováno do jaké třídy (vertikální osa) a jaká třída doopravdy patřila těmto prvkům (horizontální osa). Z podstaty tedy vyplývá rovnost A=B. Jednotlivé hodnoty matice obsahují hodnotu četnosti výskytu pravdivé třídy vůči klasifikované třídě.

		Pravdivé třídy		
		Pozitivní	Negativní	
Pravděpodobnostní třídy	Pozitivní	PRAVDA (TP)	NEPRAVDA (FP)	$\frac{TP}{(TP+FP)}$
	Negativní	NEPRAVDA (FN)	PRAVDA (TN)	$\frac{TN}{(TN+FN)}$
		$\frac{TP}{(TP+FN)}$	$\frac{TN}{(TN+FP)}$	$\frac{TP+TN}{(TP+TN+FP+FN)}$

Obr. 27) Vzor Confusion matrix

Z matice lze velmi zřetelně poznat, které třídy jsou při klasifikaci zaměňovány a které jsou naopak dobře oddělitelné. Čím vyšší čísla (počet zařazení) na hlavní diagonále dostaneme, tím přesněji model prvky klasifikuje. Okolní hodnoty znázorňují „zmatení“ klasifikačního modelu – odtud název matice Confusion matrix.



Obr. 28) K-NN Confusion matrix simulovaných dat

Na Obr.28) lze vidět výslednou Confusion matici klasifikační metody K-NN pro simulovaná data. Úspěšnost na hlavní diagonále dosahuje vysokých hodnot. Mimo hlavní diagonálu se výrazné nachází pouze dvě „záměny“ a to klasifikace měření jako „nevývahaSt-2“ přičemž správně patřilo měření do třídy „nevývahaSt-1“ (v 16,2 % případů tohoto stavu měření) a naopak klasifikace měření jako „nevývahaSt-1“ přičemž správně měření patřilo do třídy „nevývahaSt-1“ (v 33,4 % tohoto stavu měření). Z výsledků lze tedy usoudit, že tyto dva typy stavů stroje jsou si nejvíce podobná. Ostatní výsledné matice záměn jsou k nahlédnutí v příloze.

V Tab 4) jsou zaznamenány výsledky úspěšnosti klasifikačních modelů pro simulovaná a naměřená data, které byli spuštěny 10x za sebou.

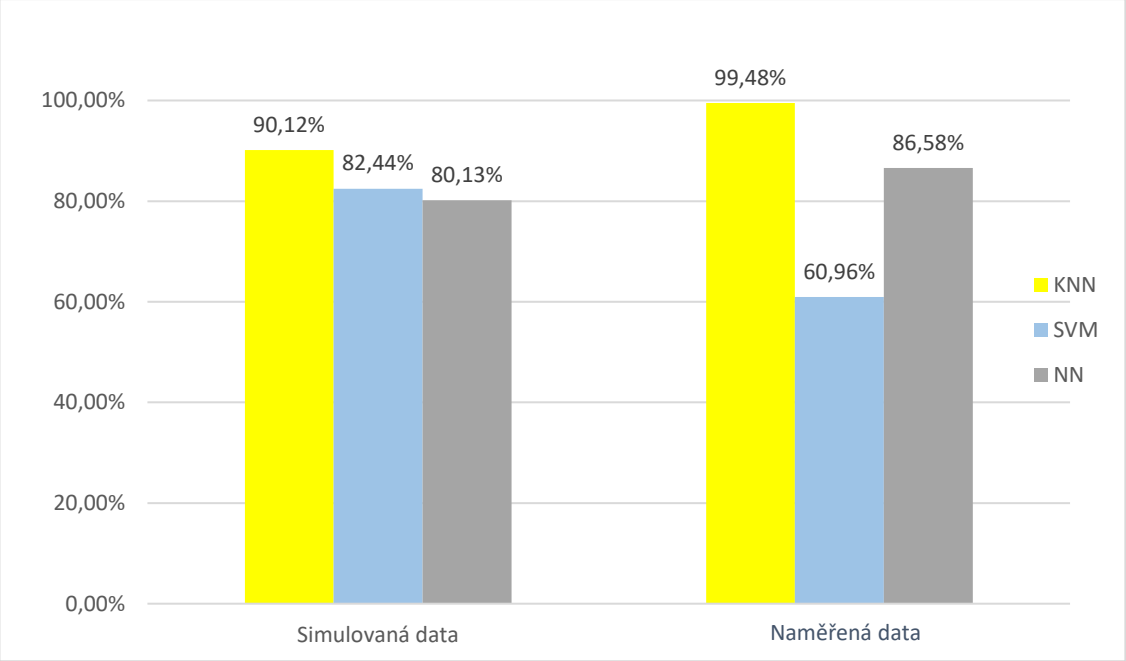
	Data					
	Simulovaná	Naměřená	Simulovaná	Naměřená	Simulovaná	Naměřená
	KNN		SVM		Neuronová síť	
1.	90,12%	99,48%	82,44%	60,96%	79,84%	80,20%
2.	90,12%	99,48%	82,44%	60,96%	79,72%	89,36%
3.	90,12%	99,48%	82,44%	60,92%	80,12%	90,20%
4.	90,12%	99,48%	82,44%	60,96%	80,16%	87%
5.	90,12%	99,48%	82,44%	62,45%	80,36%	92,32%
6.	90,12%	99,48%	82,44%	60,96%	79,32%	83,44%
7.	90,12%	99,48%	82,44%	60,96%	82%	83,68%
8.	90,12%	98,02%	82,44%	61,93%	79,92%	89,28%
9.	90,12%	99,48%	82,44%	60,96%	80%	86,80%
10.	90,12%	99,48%	82,44%	60,96%	79,88%	83,48%
Průměr	**90,12%**	**99,33%**	**82,44%**	**61,20%**	**80,13%**	**86,58%**
Směrodatná odchylka	0,000	0,005	0,000	0,005	0,007	0,038

Tab 4) Výsledky klasifikace pro jednotlivé algoritmy

Během testování se ukázalo, že metoda K- nejbližších sousedů je pro oba soubory dat nejvhodnější, jelikož dosahuje vysoké úspěšnosti bez výrazného „zmatení“.

U mého měření je směrodatná odchylka u KNN a SVM rovna 0 z důvodu, že k promíchání učících a testovacích množin dat dochází pouze ve skriptu při zpracování dat a poté si modely při každém učení/testování modelu tahají stejnou množinu dat.

KNN a SVM jsou oproti neuronové síti při klasifikačních úkolech v podstatě deterministické, proto při stejné učící a testovací množině budou vykazovat opakovaně stejné výsledky. Stejně tomu ale není u neuronové sítě, kde opakované učení často výrazně změní výsledek testování i při zachování učící a testovací množiny.



Graf 1)Spolehlivost klasifikovaných modelů

Metoda SVM projevovala vyšší nepřesnost zejména u reálných dat, proto se jevila jako nejméně vhodná. Výsledky modelu neuronové sítě pak velmi záviseli na době učení a maximálním počtu iterací. Tím, že je navíc učení nedeterministické (se stochastickými vlastnostmi) se míra nepřesnosti lišila při opakovaném přeučování.

6 ZÁVĚR

Cílem mé diplomové práce byla nejprve rešerše v oblasti metod umělé inteligence, použití umělé inteligence v technické diagnostice, výběr a popis SW prostředí a klasifikačních metod pro vyhodnocení poruch na rotačním stroji. Hlavním úkolem této práce byl vlastní návrh řešení, kde byli zpracovány klasifikačních modelů, které mohou být v budoucnu implementovány na různé soubory dat. Práce obsahuje přílohu se zpracovaným Manuálem pro studenty nebo osoby, které budou s danými skripty a modely v budoucnu pracovat.

Pro zpracování dat jsem si zvolila SW prostředí GNU Octave z licenčních důvodů. Popsala jsem celkový postup při vytváření klasifikačních modelů a jakým způsobem jednotlivé modely pracují. Nejprve byla popsána zdrojová data a jejich zpracování, aby bylo možné implementovat klasifikační modely. Vyhodnocením klasifikačních modelů je Confusion matrix, které jsou poté mezi sebou porovnávány.

V praktické části je provedena klasifikace dat pomocí tří klasifikátorů, které jsem si zvolila. Existuje velké množství klasifikačních modelů, které lze na soubory dat implementovat. Jedná se především o klasifikátory K-NN, SVM a mnou specifikovanou neuronovou síť.

Nejprve klasifikaci předcházelo zpracování dvou souborů dat, jednalo se o data simulovaná pro otestování metodiky a data získaná z reálného měření. Simulovaná data byla vygenerována na základě konkrétních poruch na rotačních strojích v simulačním softwaru. Reálná data byla naměřena pomocí analogového akcelometru na rotačním stroji v ose X a Y při statické a dynamické nevývaze. Dále z každého typu dat byla náhodně vybrána polovina dat pro učení a druhá pro otestování, abych byla schopná otestovat správnost klasifikačních modelů.

Před implementací klasifikátoru bylo nutné si data předpřipravit. Úkolem bylo především vybrat hodnoty dat, které jsou podstatné pro klasifikaci a zpracovat je do vhodného tvaru pro implementaci klasifikačních modelů. Vybrané hodnoty jsou tzv. prediktory, které každé měření shrnují do několika charakteristických hodnot. Zvolila jsem prediktory RMS, STDV a FFT. Výsledkem předzpracování souborů dat byly matice o velikosti 500x14 pro každou třídu měření (naměřená data, simulovaná data). Jednotlivé sloupce matice představují prediktory RMS, STDV a FFT a fázový posun mezi stojany v obou osách X a Y, který reprezentuje zpoždění mezi signály. Řádky v matici představují jednotlivá měření.

Následně byla provedena implementace klasifikačních modelů, kde musí být data nejprve natrénovány na učící množině a poté otestovány na testovací množině. Učící model se skládá ze tří základních kroků, které jsou inicializace, učení a uložení modelu.

Jednotlivé klasifikační modely jsou mezi sebou porovnávány pomocí Confusion matrix (matice záměn). Jedná se o matici, která vyobrazuje kompletní přehled klasifikace prvků do tříd a přesnost modelu. V maticích je zřetelně poznat, které třídy během klasifikace jsou zaměňovány a které jsou naopak od sebe dobře oddělitelné. Čím vyšší čísla na hlavní diagonále dostaneme, tím přesněji model prvky klasifikuje. Okolní hodnoty představují tzv. „zmatení“ klasifikačního modelu.

Ověření přesnosti klasifikačních modelů pro testovací a naměřená data bylo provedeno po opakovaném spuštění klasifikačních modelů. Během testování klasifikačních modelů se ukázalo, že model KNN se jeví pro oba soubory dat jako nejvhodnější, jelikož dosahuje vysoké přesnosti bez výrazného zmatení. Směrodatná odchylka u modelu KNN a SVM je nulová,

jelikož promíchání učících a testovací dat probíhá pouze ve skriptu při zpracování dat a klasifikační modely při každém učení/ testování modelu tahají stejnou množinu dat.

Modely KNN a SVM jsou v podstatě deterministické, proto při stejné učící a testovací množině vykazují stejné výsledky. Neuronová síť závisí především na době učení a maximálním počtu iterací. Jedná se o nedeterministické učení se stochastickými vlastnostmi, jehož míra nepřesnosti je způsobena opakovatelným přeučováním.

V budoucnu by bylo možné implementovat další klasifikační modely na soubory dat. Spolehlivost modelů by byla možná ověřit i jiným způsobem než srovnávací metodou pomocí Confusion matrix. Závěrem mohu konstatovat, že veškeré cíle mé diplomové práce se mi podařilo naplnit.

7 SEZNAM POUŽITÝCH ZDROJŮ

[1] Stručné dějiny umělé inteligence: jak vznikala a co nás čeká? *https://zoom.iprima.cz/* [online]. 17.3.2016 [cit. 2021-04-19]. Dostupné z: https://zoommagazin.iprima.cz/veda/strucne-dejiny-umele-inteligence-jak-vznikala-co-nas-ceka

[2] VACEK, Ludvík. Historie a současnost umělé inteligence. *https://www.fi.muni.cz/usr/jkucera/pv109/2000/: Studentské práce z roku 2000* [online]. 2000 [cit. 2021-04-19]. Dostupné z: https://www.fi.muni.cz/usr/jkucera/pv109/2000/xvacek.htm

[3] HAMMER, Miloš. *Metody umělé inteligence v diagnostice elektrických strojů*. 1.vyd. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-231-2.

[4] HÁJEK, Petr. *Metamathematics of Fuzzy Logic*, Praha: Springer, 2001

[5] VONDRÁK, Ivo. *Umělá inteligence a neuronové sítě*. 3.vyd. Ostrava: VŠB-TUO, 2009. ISBN ISBN 978-80-248-1981-5.

[6] Co je strojové učení? *Oracle* [online]. Oracle, © 2021 [cit. 2021-04-19]. Dostupné z: https://www.oracle.com/cz/data-science/machine-learning/what-is-machine-learning/

[7] Co je machine learning? *Microsoft Azure* [online]. Microsoft, © 2021 [cit. 2021-04-19]. Dostupné z: https://azure.microsoft.com/cs-cz/overview/what-is-machine-learning-platform/

[8] What Is Machine Learning?? A Key For The Beginners. *SEOMAG* [online]. 22.7.2019, [cit. 2021-04-19]. Dostupné z: https://bigdataevo.blogspot.com/2019/07/what-is-machine-learning-key-for.html?fbclid=IwAR1Mion6QZdM-BCrShTlCf7I7990fQXK2JVXJUWCPt9N6n_wAhD694drnJ8

[9] ŠÍMA, Jiří a Roman NERUDA. *Teoretické otázky neuronových sítí*. 1. vyd. Praha: MATFYZPRESS 1996. 196 s. ISBN 80-85863-18-9.

[10] Expertní systémy. *https://is.mendelu.cz/* [online]. [cit. 2021-04-19]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21856

[11] KREIDL, Marcel a Radislav ŠMÍD. *Technická diagnostika: senzory, metody, analýza signálu*. Praha: BEN – technická literatura, 2006. ISBN 80-730-0158-6

[12] Stavový prostor. *http://www.fit.vutbr.cz/* [online]. 30.5.1999 [cit. 2021-04-20]. Dostupné z: http://www.fit.vutbr.cz/~tisnovpa/publikace/diplomka/doc/node32.html

[13] Stavový prostor a jeho prohledávání. *https://cw.fel.cvut.cz/* [online]. [cit. 2021-04-20]. Dostupné z: https://cw.fel.cvut.cz/old/_media/courses/y33zui/01_neinformprohled_v2.pdf

[14] Hluboké učení. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-04-20]. Dostupné z: https://cs.wikipedia.org/wiki/Hlubok%C3%A9_u%C4%8Den%C3%AD

[15] Umělá inteligence a učení. *EdTech KISK* [online]. 2019, 29.5.2019 [cit. 2021-04-20]. Dostupné z: https://medium.com/edtech-kisk/um%C4%9Bl%C3%A1-inteligence-a-u%C4%8Den%C3%AD-9be6c0e0d78d

[16] SCHWARZ, D. a KAŠPÁREK, T. 2007, 'Brain tissue classification with automated generation of training data improved by deformable registration', *Computer Science*, vol. 4673, pp. 301-303

[17] Information retrieval. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Information_retrieval

[18] Text mining. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Text_mining

[19] Question answering. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Question_answering

[20] Machine translation. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Machine_translation

[21] Automated planning and scheduling. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Automated_planning_and_scheduling

[22] VOLNÁ, Eva. *Neuronové sítě 1* [online]. 2. Ostravská univerzita v Ostravě, 2008 [cit. 2021-4-26]. Dostupné z: https://web.osu.cz/~Volna/Neuronove_site_skripta.pdf

[23] Reinforcement learning. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Reinforcement_learningh

[24] Combinatorial optimization. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Combinatorial_optimization

[25] Natural language processing. *Wiki* [online]. [cit. 2021-4-26]. Dostupné z: https://cs.qwe.wiki/wiki/Artificial_intelligence#Natural_language_processing

[26] HLAVÁČ, Václav. *Úvod do robotiky* [online]. Praha: Fakulta elektrotechnická ČVUT v Praze katedra kybernetiky, Centrum strojového vnímá [cit. 2021-4-26]. Dostupné z: http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/51Robotika/01UvodRobotika.pdf

[27] Text to Speech (TTS). *Techopedia* [online]. © 2021 [cit. 2021-04-19]. Dostupné z: https://www.techopedia.com/definition/23843/text-to-speech-tts

[28] Co je umělá inteligence – AI? *Oracle* [online]. © 2021 [cit. 2021-04-20]. Dostupné z: https://www.oracle.com/cz/artificial-intelligence/what-is-ai/#what-is-ai

[29] *Umělá inteligence* [online]. Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity [cit. 2021-4-26]. Dostupné z: https://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--vicerozmerne-metody-pro-analyzu-dat

[30] HOLČÍK, Jiří. *Analýza a klasifikace dat* [online]. Brno: Akademické nakladatelství CERM, s.r.o, 2012 [cit. 2021-4-26]. ISBN 978-80-7204-793-2. Dostupné z: https://www.matematickabiologie.cz/res/file/ucebnice/holcik-analyza-klasifikace-dat.pdf

[31] KORIŤÁKOVÁ, Eva. *Analýza a klasifikace dat* [online]. 2016 [cit. 2021-4-26]. Dostupné z: https://is.muni.cz/el/1431/podzim2016/Bi0034/um/AKD_prednaska_05_FLDA.pdf

[32] DVOŘÁK, Jiří. *Expertní systémy* [online]. Brno: VUT- Ústav automatizace a informatiky, 2004 [cit. 2021-4-27]. Dostupné z: http://www.uai.fme.vutbr.cz/~jdvorak/Opory/ExpertniSystemy.pdf

[33] PIJÁK, Marek. *KLASIFIKÁTOR EMAILOVÉ KOMUNIKACE* [online]. Brno, 2018 [cit. 2021-04-20]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=180892. Diplomová práce. VUT.

[34] Teorie chaosu. *Wikipedie: Otevřená encyklopedie* [online]. 11. 11. 2020 [cit. 2021-04-19]. Dostupné z: https://cs.wikipedia.org/wiki/Teorie_chaosu

[35] JANDA, Ondřej. *UMĚLÁ INTELIGENCE V DIAGNOSTICE VÝKONOVÝCH OLEJOVÝCH TRANSFORMÁTORŮ* [online]. Brno, 2012 [cit. 2021-04-20]. Dostupné z:

https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=60574. Disertační práce. VUT

[36] Marsland, S.: Machine Learning, An Algorithmic Perspective. New York: CPC Press, Taylor & Francis Group, první vydání, 2009, 383 s., ISBN 978-1-4200-6718-7

[37] Decision tree. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-04-20]. Dostupné z: https://en.wikipedia.org/wiki/Decision_tree

[38] STREJC, Petr. *Shluková analýza ve financích* [online]. Praha, 2009 [cit. 2021-04-20]. Dostupné z: https://dspace.cuni.cz/bitstream/handle/20.500.11956/28133/BPTX_2008_1_11320_0_232614_0_66669.pdf?sequence=1&isAllowed=y. Bakalářská práce. UK.

[39] John W. Eaton, S. H., David Bateman; Wehbring, R.: GNU Octave version 5.0.1 manual: a high-level interactive language for numerical computations. CreateSpace Independent Publishing Platform, 2018, ISBN 1441413006. http://www.gnu.org/software/octave/doc/interpreter

[40] Python. *Wikipedie: Otevřená encyklopedie* [online]. [cit. 2021-04-19]. Dostupné z: https://cs.wikipedia.org/wiki/Python

[41] Strojové učení a software Matlab. *Automa* [online]. 2015, **9/2015**(8), 82-84 [cit. 2021-04-20]. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/54029.pdf

[42] *Matlab Documentation* [online]. The MathWorks, 2021 [cit. 2021-4-26]. Dostupné z: https://www.mathworks.com/help/matlab/

[43] *Knime: Open for Innovation* [online]. 2021 [cit. 2021-4-26]. Dostupné z: https://www.knime.com/

[44] PLATT, John C. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research Technical Report MSR-TR98-14, 1998. Dostupné z: http://research.microsoft.com/ jplatt/smoTR.pdf

[45] ZUTH, Daniel a MARADA, Tomáš. Using artificial intelligence to determine the type of rotary machine. *Mendel computing journal* [online]. **2018**(2), 49-54 [cit. 2021-4-26]. ISSN 1803-3814. Dostupné z: https://mendel-journal.org/index.php/mendel/article/view/10/6

[46] *(Diskrétní) Fourierova transformace* [online]. Olomouc: UPOL, 2003 [cit. 2021-4-26]. Dostupné z: http://apfyz.upol.cz/ucebnice/down/mini/fourtrans.pdf

[47] CHIH-CHUNG, Chang a Lin CHIH-JEN. LIBSVM - A Library for Support Vector Machines. *http://www. csie.ntu.edu.tw* [online]. [cit. 2021-4-26]. Dostupné z: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[48] PRIDDY, Kevin L. a Paul E. KELLER. *Artificial neural networks: an introduction*. 2005. Bellingham, Wash: SPIE Press. ISBN 978-0819459879.

[49] BURGET, Radim. Dolování znalostí z báze dat. *Teoretická informatika* [online]. Brno: Ústav telekomunikací, Vysoké učení technické v Brně, 3.11.2010, [cit. 2021-04-26]. Dostupné z: https://wwww.vutbr.cz/elearning/mod/resource/view.php?id=114594

[50] Genetické algoritmy. *VSB* [online]. [cit. 2021-4-26]. Dostupné z: https://poli.cs.vsb.cz/edu/isy/down/ga.

[51] S. RAJASEKARAN a V. P. G.A. „Neural Network, Fuzzy Logic, and Genetic Algorithms - Synthesis and Application. *Prentice Hall*. 2005, 157-186

[52] Richard O. Duda, Peter E. Hart, David G. Stor. *Pattern Classification*. Wiley, 2001. ISBN 0471056693.

[53] Hyperplane and support vectors. *ResearchGate* [online]. [cit. 2021-4-27]. Dostupné z: https://www.researchgate.net/figure/Hyperplane-and-support-vectors-Vectors-of-class-A-are-in-red-and-vectors-of-class-B-are_fig4_224545094

8 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

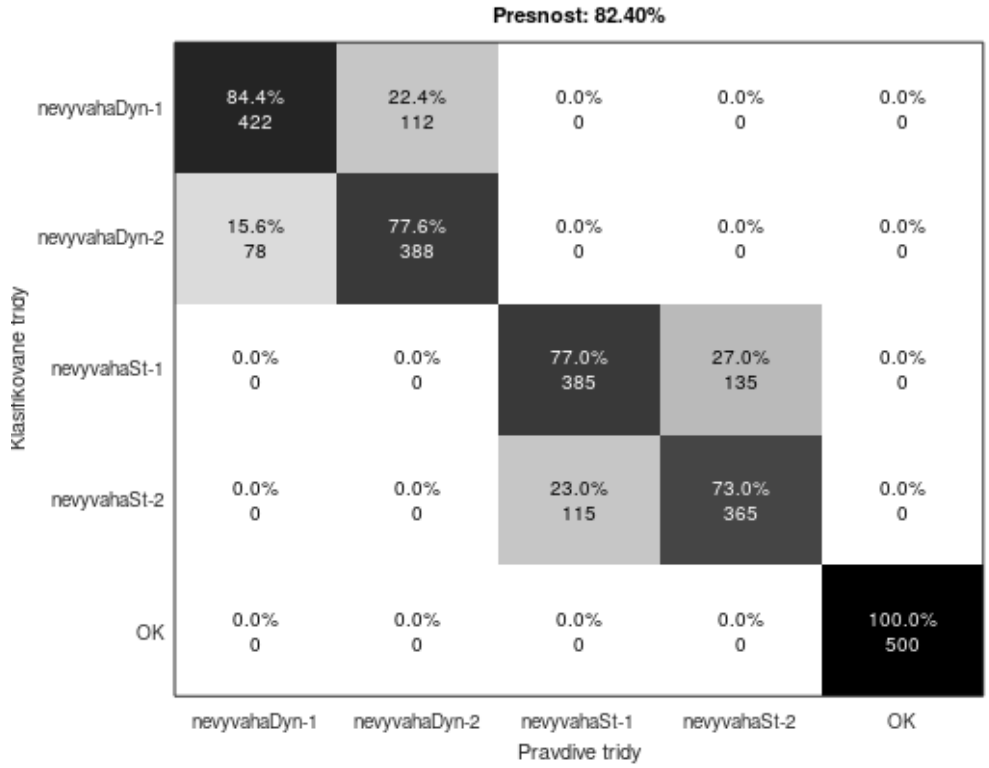
8.1 Seznam tabulek

8.2 Seznam obrázků

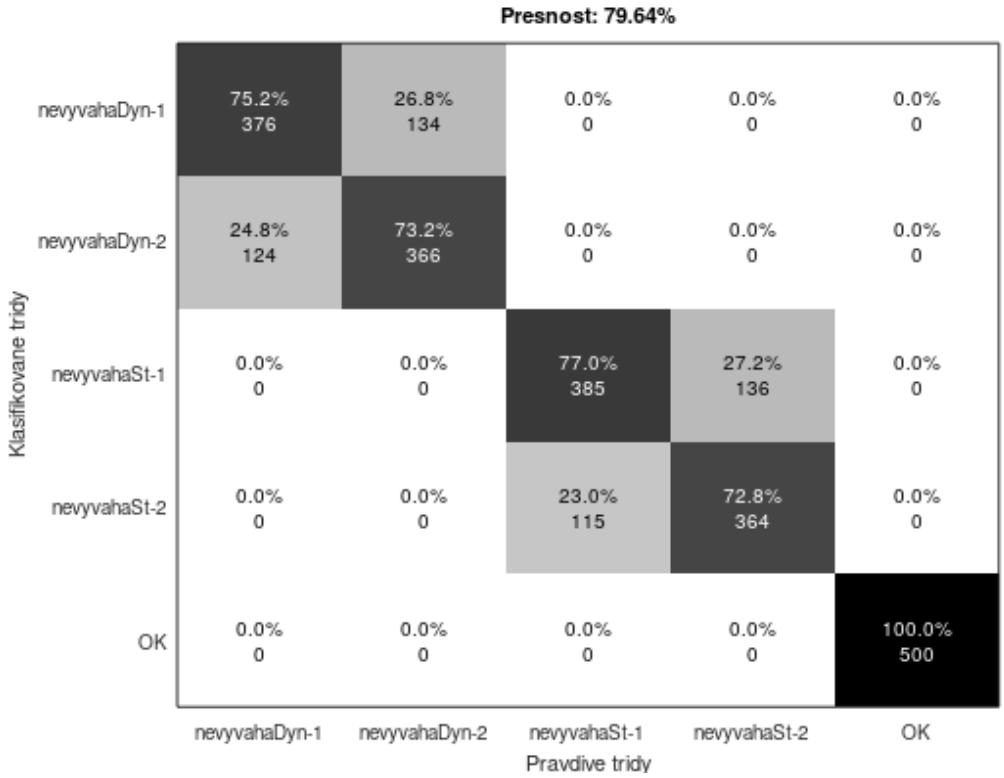
9 SEZNAM PŘÍLOH

PŘÍLOHA 1

Confusion matrix pro simulovaná data



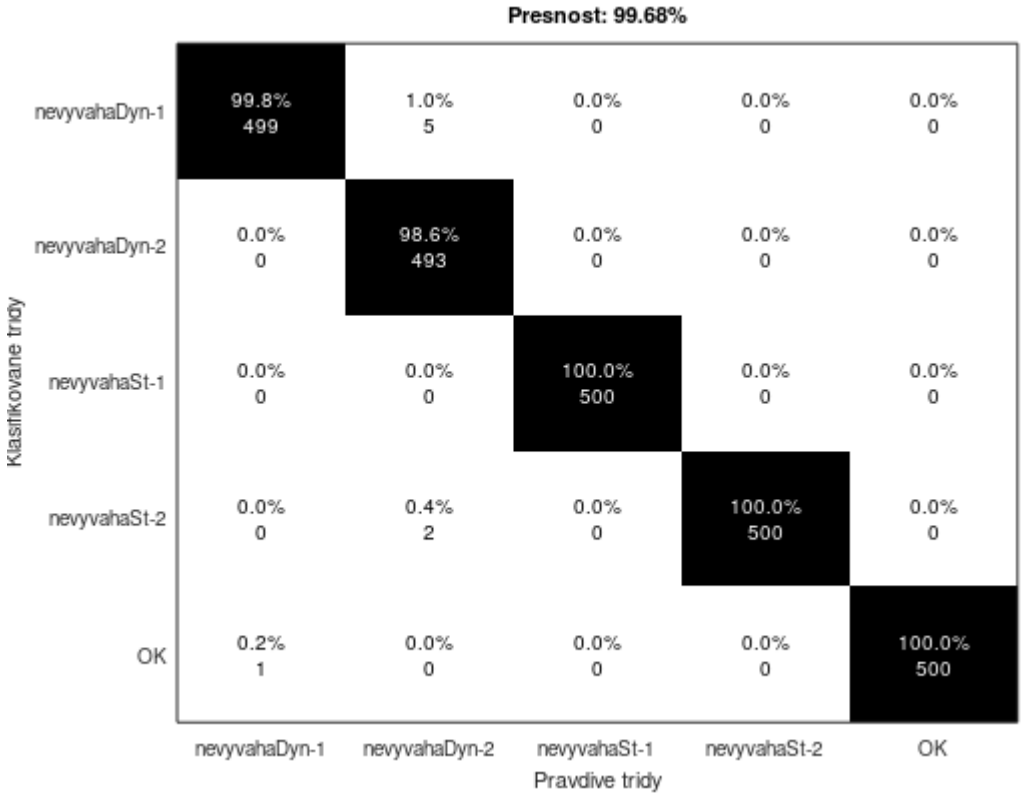
Obr. 29) Confusion matrix SVM pro simulovaná data



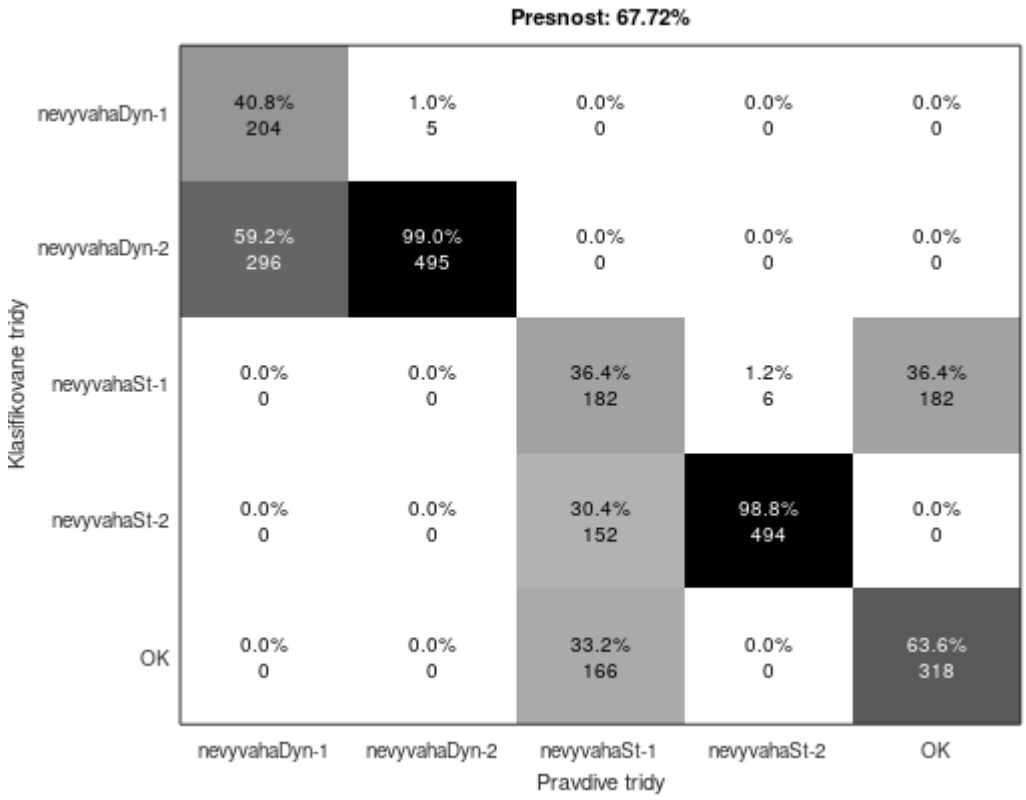
Obr. 30) Confusion matrix NN pro simulovaná data

PŘÍLOHA 2

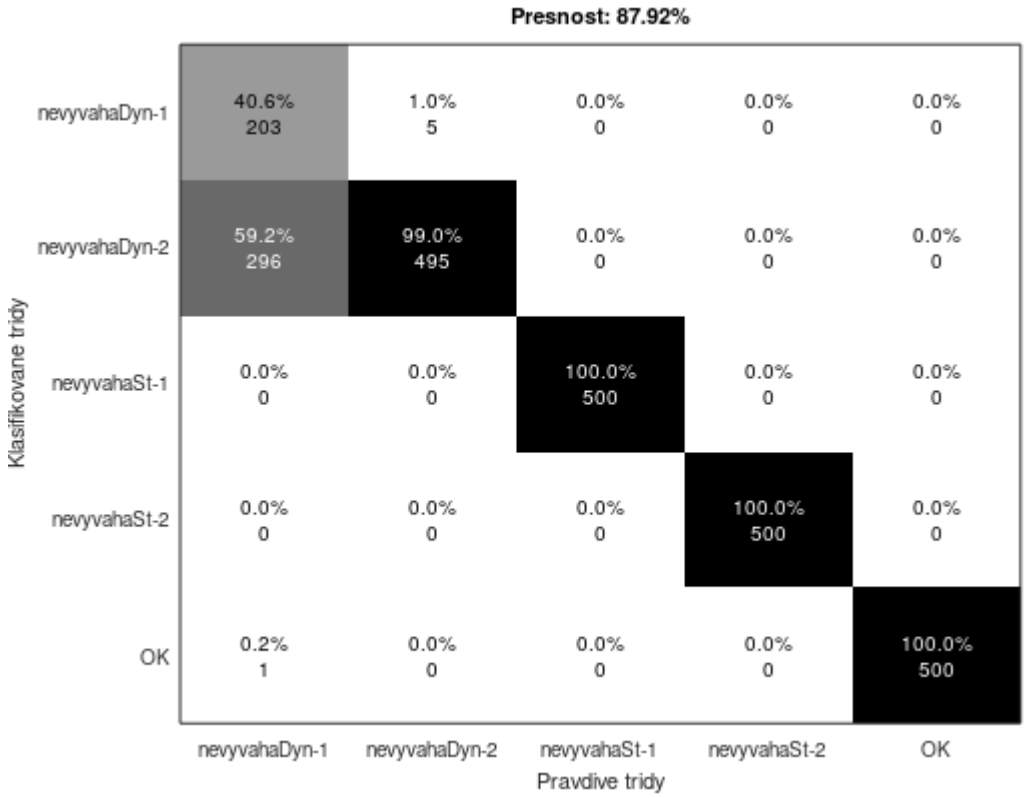
Confusion matrix pro naměřená data



Obr. 31) Confusion matrix KNN pro naměřená data



Obr. 32) Confusion matrix SVM pro naměřená data

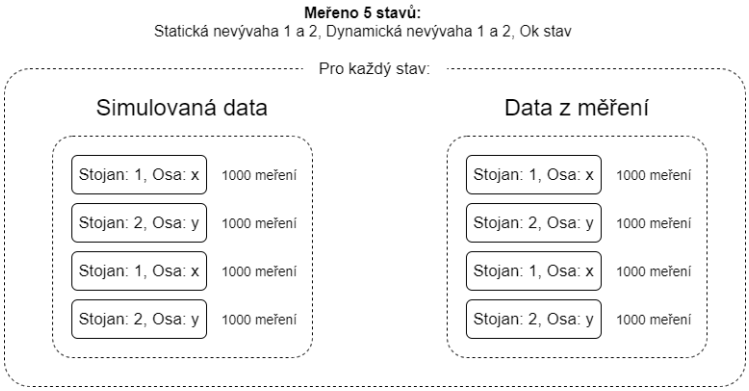


Obr. 33) Confusion matrix NN pro naměřená data

PŘÍLOHA 3

Manuál pro studenty

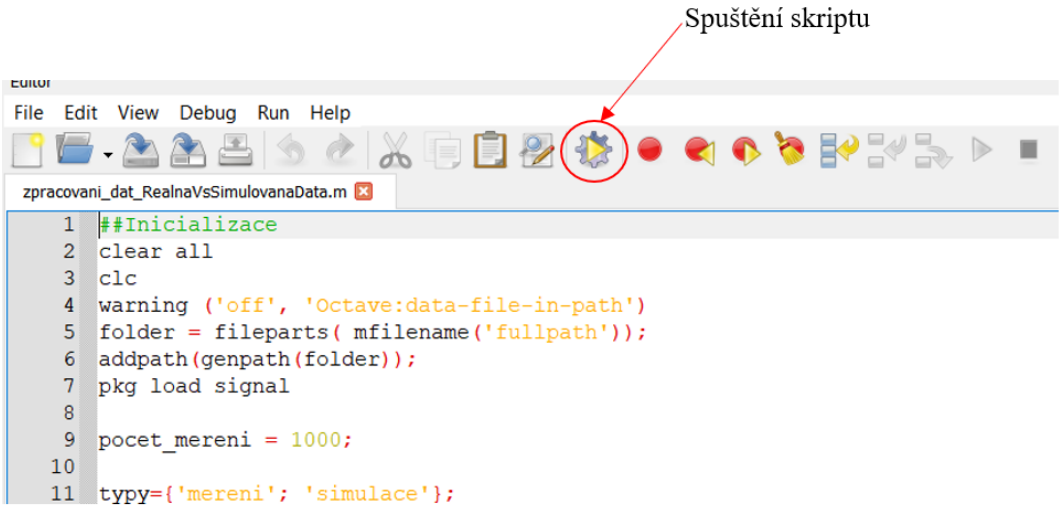
Pro práci se skripty je nutné mít nainstalovaný program GNU Octave. Vstupní daty jsou dva soubory dat, které jsou data simulovaná pro otestování metodiky a data získaná z reálného měření. Na obr. 1 je znázorněna struktura dat.



Obr.1) Struktura dat

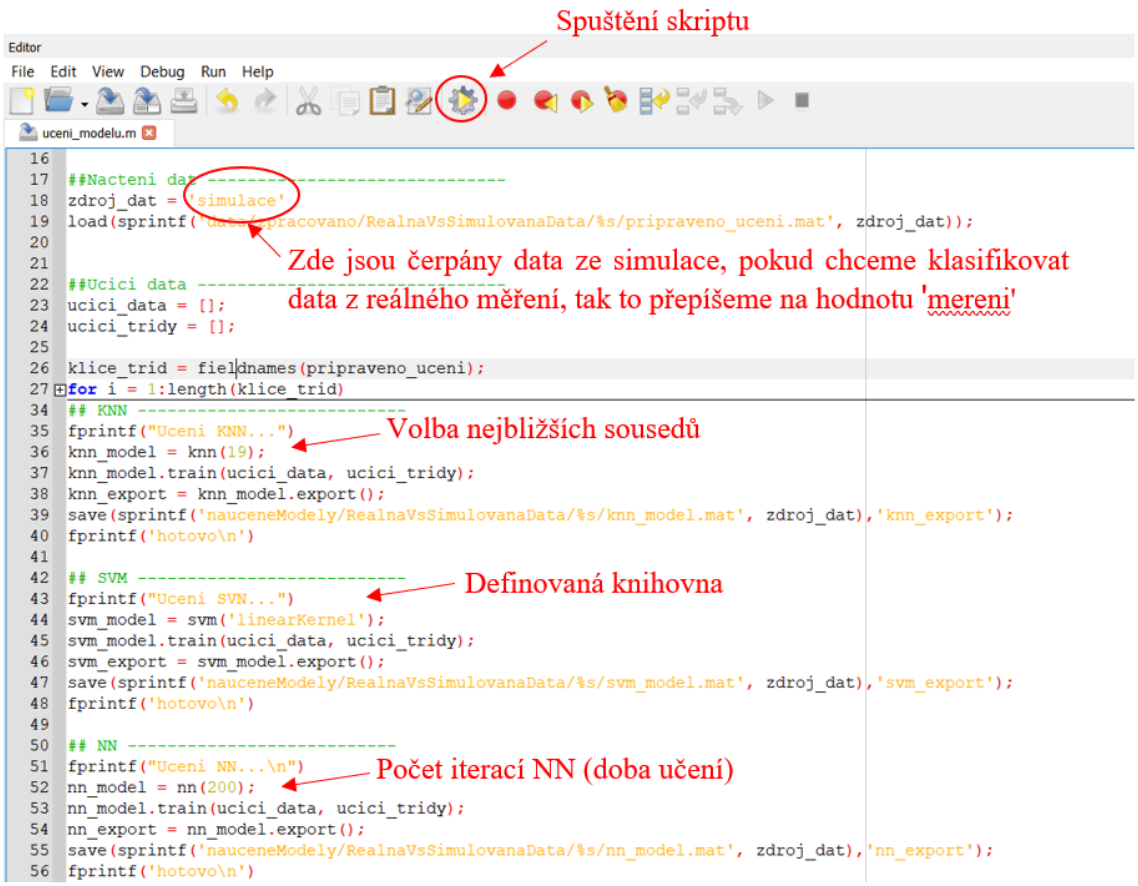
Při práci s jinými daty, je nutné přepsat a roztřídit si data. Dále ve skriptu přepsat názvy souborů, aby věděl skript odkud si má data tahat.

1. **Otevřete si program GNU Octave (GUI)**
2. **Otevřete si skript zpracováni_dat_RealnaVsSimulovanaData.m**
    1. Zde probíhá předpříprava dat. Cílem je vybrat hodnoty, které jsou podstatné pro klasifikaci a zpracovat do tvaru, který je vhodný pro klasifikační modely
    2. Probíhá zde míchání dat pro trénování a pro otestování, aby byla ověřena správnost modelu.
    3. Prediktory, které jsou ve skriptu jsou FTT, RMS a STDV



- Výsledkem je 5 matic o velikosti 500x14 ((RMS_1X, STDV_1X, FFT_1X, RMS_2X, STDV_2X, FFT_2X, FAZE_X, RMS_1Y, STDV_1Y, FFT_1Y, RMS_2Y, STDV_2Y, FFT_2Y, FAZE_Y)
- Velikost a struktura matice se může lišit, pokud změníme soubory dat pro předpřípravu.

3. **Otevřete si skript uceni_modelu.m**
    1. Zde probíhá implementace klasifikačních modelů (K-NN, SVM a neuronová síť)
    2. Modely musí být nejprve pro klasifikaci dat natrénovány na učící množině dat a poté otestovány na množině testovací



4. **Otevřete si skript test_modelu.m**
    1. Opět spustíme skript stejným způsobem jako v předchozích skriptech
    2. Zde probíhá otestování klasifikačního modelu, kde výsledkem jsou confusion matice pro K-NN, SVM a NN
    3. KNN a SVM jsou metody deterministické, které při stejné učící a testovací množině vykazují stejné výsledky.
    4. U neuronové sítě, při opakovaném učení se často mění výsledek testování i při zachování učící a testovací množiny.
    5. Neuronová síť je nedeterministické učení (nepřesnost je způsobená při opakovaném přeučováním).